Visual Studio 2003 Visual J#

Copyright© 2016 Microsoft Corporation

The content in this document is retired and is no longer updated or supported. Some links might not work. Retired content represents the latest updated version of this content.

Visual Studio .NET

Microsoft Visual Studio.net	Microsoft® Visual Studio® .NET is the comprehensive tool set for rapidly building and integrating XML Web services, Microsoft Windows®–based applications, and Web solutions. Today, the world takes a little step into the future.

Check out the Visual Studio .NET Developer Center

You'll find in-depth content, community resources, downloads, product information, and more at this one-stop resource for Visual Studio .NET developers.

In This Library Section	Essentials
• **Documentation** • **Technical Articles** • **Introducing Visual Studio .NET** • **Developing with Visual Studio .NET** • **Visual Studio Walkthroughs** • **Code Samples**	• **Frequently Asked Questions About Visual Studio .NET** • **Inside the .NET Framework** • **Upgrading to .NET** • **Visual Studio .NET Support Center** • **Newsgroups** • **Communities**

Walkthrough: Handling a Concurrency Exception with Visual J#

In this walkthrough you will create a Windows application that raises a concurrency error and illustrates one strategy for handling it. The walkthrough simulates two users working with the same data at the same time. The Windows Form you will create allows you to act as both users from a single form.

This walkthrough will demonstrate the following tasks:

1. User 1 and User 2 fill their respective datasets with the same data.
2. User 2 edits a record, updates the dataset, and writes the changes to the data source.
3. User 1 edits the same record, updates the dataset, and attempts to write the changes to the data source, which results in a concurrency error being raised.

You will catch the error, then display the different versions of the record, allowing the user (you) to determine what should happen with the pending changes made by User 1.

> **Note** Using a dataset is only one option for data access, and is not the optimal choice in some scenarios. Nonetheless, datasets are usually the right choice in Windows Forms applications, and this walkthrough illustrates one scenario in which datasets are an appropriate choice. For more information, see Recommendations for Data Access Strategies.

Prerequisites for this walkthrough:

- Access to the Pubs SQL Server sample database with permission to perform updates.

Create a New Project and Data Connection

You begin your walkthrough by creating a new Windows application in Visual J#.

To create a new project

1. From the **File** menu, point to **New**, and then click **Project** to display the **New Project** dialog box.
2. Select **Visual J# Projects** in the **Project Types** pane, and then select **Windows Application**.
3. Name the project **concurrency_walkthrough**, then click **OK**.

   Visual Studio will add the concurrency_walkthrough project to Solution Explorer and display a new Windows Form in the designer.

To create a new data connection

1. In Server Explorer, create a new connection to the Pubs sample database. For more information, see Adding New Data Connections in Server Explorer.
2. In Server Explorer, expand the connection you created in the previous step.
3. Expand the **Tables** area.
4. Drag the **authors** table onto your form.

   A **Connection** object and a **DataAdapter** object appear in the component tray below the form.

Create the Datasets

In this section you will create two datasets named **DsAuthors1** and **DsAuthors2**. These datasets will represent the data that the two simultaneous users are working with. Next you will add two DataGrid controls to the form and bind them to the datasets. Finally, two Button controls will be added to the form: an **Update** button and a **Reset** button. The **Update** button will change a record in **DsAuthors1** and attempt to write the change to the database. The **Reset** button resets the database back to the original record so the walkthrough will work if run more than once.

To create two new datasets

1. Select the **DataAdapter** object.
2. From the **Data** menu, select **Generate Dataset**.

   The **Generate Dataset** dialog box appears.

3. Select **New** and name the dataset **DsAuthors**.

An instance named **DsAuthors1** appears in the component tray.

4. From the **Data** tab of the Toolbox, drag a **DataSet** onto the form.

   The **Add Dataset** dialog box appears.

5. Confirm that **Typed dataset** is selected and **concurrency_walkthrough.DsAuthors** appears in the **Name** box.

   An instance named **DsAuthors2** appears in the component design tray.

Bind the Data and Add Buttons

The data grids are used only to display the data. No data should be edited in the data grids during this walkthrough.

To add two DataGrids to the form

1. From the **Windows Forms** tab of the Toolbox, drag a **DataGrid** object onto the left side of your form.
2. From the **Windows Forms** tab of the Toolbox, drag a **DataGrid** object onto the right side of your form.

To bind the datasets to the DataGrid Controls

1. Select **DataGrid1** and set the following properties in the **Properties** window:

Property	Setting
DataSource	**DsAuthors1**
DataMember	**authors**
CaptionText	**DsAuthors1**

2. Select **DataGrid2** and set the following properties in the **Properties** window:

Property	Setting
DataSource	**DsAuthors2**
DataMember	**Authors**
CaptionText	**DsAuthors2**

To add the Update and Reset buttons to the form

1. From the **Windows Forms** tab of the Toolbox, drag a **Button** control onto the form and place it above **DataGrid1**.
2. Select the button and then, in the **Properties** window, name the button **btnUpdate** and set its **Text** property to **Update**.
3. From the **Windows Forms** tab of the Toolbox, drag a second **Button** object onto the form and place it above **DataGrid2**.
4. Select the button and then, in the **Properties** window, name the button **btnReset** and set its **Text** property to **Reset**.

Reset the Database

You will add code to the form that will reset the database to known values (in case you want to run the walkthrough more than once).

To add code to reset the database

- Right-click the form, choose **View Code** from the shortcut menu, and then insert the following code above the constructor **Form1**:

```
// Visual J#
private void resetDatabase()
{
   // Fill the dsAuthors dataset with data.
   sqlDataAdapter1.Fill(dsAuthors1);
   // Reset the au_fname in the first row to "John".
   dsAuthors1.get_authors().get_Item(0).set_au_fname("John");
   // Write the record back to the database.
   sqlDataAdapter1.Update(dsAuthors1);
   // Close the connection.
sqlConnection1.Close();
}
```

Fill the Datasets

This step fills both datasets with the same data.

To add code to fill the datasets from the database

- Insert the following code into the Code Editor:

```
// Visual J#
private void fillthe DataSets()
{
   sqlDataAdapter1.Fill(dsAuthors1);
   sqlDataAdapter1.Fill(dsAuthors2);
   sqlConnection1.Close();
}
```

Simulate Changes by User 2

To add code to simulate changes by User 2

- Insert the following code into the Code Editor:

```
// Visual J#
private void user2changes()
{
   // Simulate a second user changing a record.
   dsAuthors2.get_authors().get_Item(0).set_au_fname("User 2");
   // Write it back to the database.
   sqlDataAdapter1.Update(dsAuthors2.GetChanges());
   // Refresh dsAuthors2 with the updated data.
   sqlDataAdapter1.Fill(dsAuthors2);
// Close the connection..
sqlConnection1.Close();
}
```

Create the Form_Load Event Handler

To add code into the Form_Load event to initialize the walkthrough

1. In Design view, double-click an empty area of the form to automatically create the **Form_Load** event handler.
2. Add code so the event handler looks like the following:

```
// Visual J#
private void Form1_Load(Object sender, System.EventArgs e)
{
   resetDatabase();
   fillthe DataSets();
   user2changes();
}
```

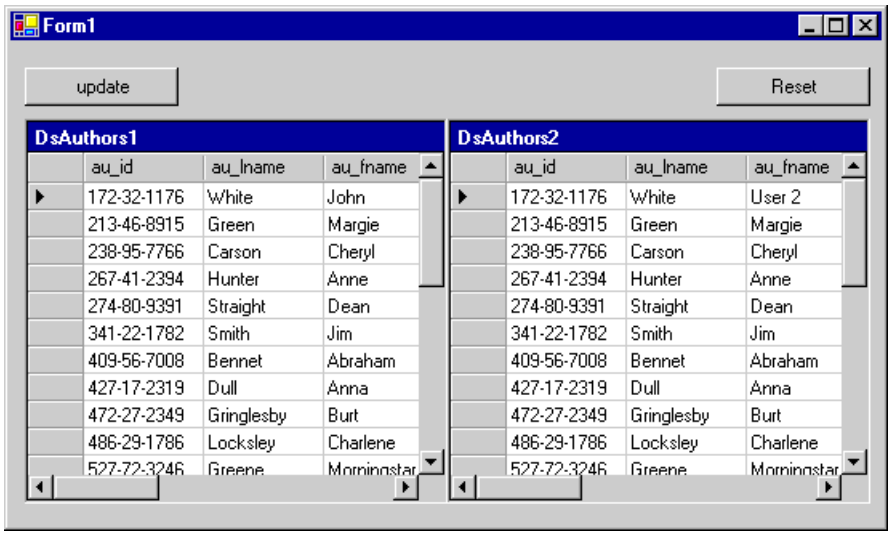
3. Save your project.

Run the Application

1. Press F5 to run the application

   The form appears with two datagrids filled with data from the authors table in the Pubs database. The **au_fname** field of the first record in **DsAuthors1** should be **John**. The **au_fname** field of the first record in **DsAuthors2** should be **User 2**.

   Your form will look similar to the following:



2. From the **Debug** menu, select **Stop Debugging**.

Update the Dataset and Write the Changes to the Database

Next, you will write code that will attempt to update the database with changes from the **DsAuthors1** dataset. If successful, the AcceptChanges method of the dataset is called and a message box displays a success message. If the update fails for any reason, the error is caught and a message box displays the error message, with the type of error object as the title of the message box. Because this walkthrough is designed to raise a concurrency error, the successful message box is shown for completeness.

> **Note** This **Try...Catch** block will catch any error. Later in this walkthrough you will add an additional catch statement to specifically handle the concurrency error.

To update the database

1. Call the Update method.
2. Create an exception handler that displays a message box.

Your code should look like this:

```
// Visual J#
private void updateDatabase()
{
   try
   {
      // Update the database with the changes from dsAuthors1.
      sqlDataAdapter1.Update(dsAuthors1.GetChanges());
      dsAuthors1.AcceptChanges();
   sqlConnection1.Close();
      MessageBox.Show("The update was successful.");
   }
   catch (System.Exception ex)
   {
      // Display information about update errors.
      MessageBox.Show(ex.get_Message(), ex.GetType().ToString());
   }
}
```

Next, you will add code that changes the au_fname column in the **DsAuthors1** dataset. The code then calls the **updateDatabase** procedure to try to write this change to the database. Because User 2 changed the value earlier, a concurrency error will be raised.

To update the DsAuthors1 dataset

1. Double-click the **Update** button.
2. Create the **btnUpdate_Click** event handler:

```
// Visual J#
private void btnUpdate_Click(Object sender, System.EventArgs e)
{
```

```
    // Change au_fname in the first row of dsAuthors1 to "User 1".
    dsAuthors1.get_authors().get_Item(0).set_au_fname("User 1");
    updateDatabase();
}
```

3. Save your project.
4. Press F5 to run the application.
5. Click the **Update** button to change the **au_fname** field of the first record to **User 1**.

   The concurrency error will be raised.

Handling Concurrency Errors

How you handle the error is dependent upon the specific business rules that govern your application. The following strategy to handle the error will be used as an illustration. The application will present the user with three versions of the record:

- The current record in the database
- The original record in the dataset
- The proposed changes in the dataset

The user will then be able to either overwrite the database with the proposed change or cancel the update and refresh the dataset.

Creating a Custom Error Handler

When you perform an update, you generally want to do so in a structured exception handler so that you can catch errors. In the code you used above, you used a **try...catch** block that caught all errors — that is, a structure that includes a generic **catch** statement for any error.

You can also catch specific errors, so that you can respond appropriately. To illustrate, this walkthrough will add an exception handler for one specific error — namely, a concurrency error, which you can examine using the DbConcurrencyException object. Here, you will handle this error by displaying information to the user.

To add specific handling for the DBConcurrencyException error

1. If the application is still running, exit run mode to return to the Code Editor.
2. Add a second **catch** statement above the existing one in the **updateDatabase** method.
3. Pass the **DBConcurrencyException** object to the **createMessage** procedure, which you will create in the next section.

```
// Visual J#
private void updateDatabase()
{
   try
   {
      // Update the database with the changes from dsAuthors1.
      sqlDataAdapter1.Update(dsAuthors1.GetChanges());
      dsAuthors1.AcceptChanges();
      MessageBox.Show("The update was successful!");
   }
   catch (DBConcurrencyException dbcx)
   {
      createMessage(dbcx);
   }
   catch (System.Exception ex)
   {
      MessageBox.Show(ex.get_Message(), ex.GetType().ToString());
   }
}
```

Displaying Choices to the User

The code you just wrote calls the **createMessage** procedure to display error information to the user. For this walkthrough, you

will use a message box to display the different versions of the record to the user and allow the user to choose whether to overwrite the record with new changes or cancel the edit.

> **Note** For simplicity's sake, this walkthrough uses the second dataset (**DsAuthors2**) as the data source for fetching the current record in the database. In a real-world application you would requery the actual data source to fetch the current value of the record that raised the error.

To create the createMessage procedure

- Create the error handler by adding the following code to the Code Editor:

```
// Visual J#
private void createMessage(DBConcurrencyException dbcx)
{
   // Declare variables to hold the row versions for display
   // in the message box.
   System.String strInDs = "Original record in dsAuthors1:\n";
   System.String strInDB = "Current record in database:\n";
   System.String strProposed = "Proposed change:\n";
   System.String strPromptText = "Do you want to overwrite the current " +
    "record in the database with the proposed change?\n";
   System.String strMessage;
   System.Windows.Forms.DialogResult response;

   // Loop through the column values.
   DataRow rowInDB = dsAuthors2.get_authors().FindByau_id(dbcx.get_Row().get_Item("Au_ID"
).ToString());
   for (int i = 0; i < dbcx.get_Row().get_ItemArray().length; i++)
   {
      strInDs = strInDs + dbcx.get_Row().get_Item(i, DataRowVersion.Original) + "\n";
      strInDB = strInDB + rowInDB.get_Item(i, DataRowVersion.Current) + "\n";
      strProposed = strProposed + dbcx.get_Row().get_Item(i, DataRowVersion.Current) + "\
n";

   }

   // Create the message box text string.
   strMessage = strInDs + "\n" + strInDB + "\n" + strProposed + "\n" + strPromptText;
   // Display the message box.
   response = MessageBox.Show(strMessage, dbcx.get_Message(),MessageBoxButtons.YesNo);
   processResponse(response);
}
```

Processing the User's Response

You will also need code to process the user's response to the message box. The options are either to overwrite the current record in the database with the proposed change or not. If the user chooses yes, the Merge method of **DsAuthors1** is called with the *preserveChanges* argument set to **true**. This takes the original versions of the data rows in **DsAuthors2** and merges them with the current versions of data rows in **DsAuthors1**. This will cause the update attempt to be successful, because the original version of the record now matches the database.

To process the user input from the message box

- Add the following code to the Code Editor:

```
// Visual J#
private void processResponse(System.Windows.Forms.DialogResult response)
{
   // Execute the appropriate code depending on the button selected
   // in the message box.
   if(response == System.Windows.Forms.DialogResult.Yes)
```

```
    {
       dsAuthors1.Merge(dsAuthors2, true);
       sqlDataAdapter1.Update(dsAuthors1);
       dsAuthors1.AcceptChanges();
    }
    else if (response == System.Windows.Forms.DialogResult.No)
    {
       dsAuthors1.Merge(dsAuthors2);
    }
 }
```

Resetting the Data

To reset the form, the **Form1_Load** method will be modified to execute when the **Reset** button is clicked.

> **Note** One method can be the event handler for multiple events of multiple objects. For more information, see Connecting Multiple Events to a Single Event Handler in Windows Forms.

To create an event handler for the btnReset button

1. Open the form in the designer.
2. Click the **Reset** button.
3. In the **Properties** window toolbar, click the **Events** button.
4. Locate the **Click** event, and then click the arrow to see all the methods that could respond to the event.

   In this case, **Form1_Load** appears because it has the right signature for a **button.click** event.

5. Select **Form1_Load**.

   The code to associate the Form1_Load method with the **btnReset.Click** event is automatically generated. **Form1_Load** now responds to both **Form1.Load** and **btnReset.Click**.

   > **Tip** To view the generated code, double-click the form and expand the dimmed "Windows Form Designer generated code" section.

Run the Application

1. Press F5 to run the application.
2. Click the **Update** button.

   The concurrency error will be raised, and the message box will appear.

See Also

Adding New Data Connections in Server Explorer | Updating the Database with a DataAdapter and the DataSet | Dataset Updates in Visual Studio .NET | ADO.NET Datasets | Concurrency Control in ADO.NET | Debugging Managed Code | Deploying Applications | Data Walkthroughs

Walkthrough: Mapping Data Source Tables to Dataset Tables with Visual J#

The structure of tables in a dataset does not always match the structure of tables in a data source. There might be several reasons for this:

- The dataset was created from an existing schema that uses different names than those in the data source.
- You want to change data element names in the dataset for convenience, readability, translation to or from a foreign language, or other reason.
- You want to control the names of typed data members when you generate a dataset from the adapter.

To enable you to match up tables and columns in the data source with those in the dataset, you can set the **TableMappings** property of a data adapter. This property contains information that establishes a correspondence between data source table columns and dataset table columns. When the adapter reads data from the data source, the **TableMappings** property is used to determine where in the dataset to write the data.

This walkthrough gives you brief overview of how to set up table mappings in a data adapter. The scenario for the walkthrough assumes that you are working with a schema provided to you by someone else. You will map the columns from a table in your database (in this case, the **Customers** table of the Northwind database) to the columns defined in the schema.

To complete this walkthrough, you will need:

- Access to a server with the Northwind SQL Server sample database.

The walkthrough is split into a number of smaller pieces:

- Creating a Windows Form. The form is used only to illustrate how to create the mapping; you could use any form or component.
- Creating a dataset with a predefined schema. The walkthrough contains a simple but complete schema that you can copy and paste into Visual Studio to simulate a schema you might have gotten from someone else.
- Creating and configuring the data adapter in which you will map the data source to the dataset.
- Adding the **DataGrid** control to show the effect of the mapping.

Creating the Project and Form

This walkthrough uses a Windows Form as the container on which you will create the data adapter and dataset. It is not necessary that you use a Windows Form — you could also use a Web Forms page or a component — but a Windows Form is easy to work with and allows you later to use a **DataGrid** control to see the effect of the mappings.

To create the project and form

1. From the **File** menu, point to **New**, and then choose **Project**.
2. In the **Project Types** pane, choose **Visual J# Projects**, and then in the **Templates** pane, choose **Windows Application**.
3. Assign a name to the project that will be unique and will conform to the naming conventions you use. For example, you might name this project **Walkthrough_Mappings**. After you have assigned a name, click **OK**.

   Visual Studio creates a new project and displays a new form in the Windows Form Designer.

Creating the Schema and Dataset

For the walkthrough, imagine that someone has provided you with a schema that defines a Customers table. The schema defines a **Customers** table that is slightly different from the one in your Northwind database.

In this section, you will add a schema in your project. The actual schema layout (the XML that defines the schema) is provided below. By default, the XML Designer generates a class file based on the schema and derived from the DataSet class. The resulting class file defines the typed dataset class that you use in your application.

For more information about datasets and how they are defined, see Introduction to Datasets.

To add the schema

1. In Solution Explorer, right-click the name of your project, choose **Add**, and then choose **Add New Item**.
2. In the **Templates** pane, choose **DataSet**, name the new item **dsCustomers**, and then click **Open**.

The XML Designer opens with a new, blank dataset schema.

3. Click the **XML** tab at the bottom.
4. Copy the following schema text and then paste it into the XML view of the designer, completely overwriting the schema already in the file.

   Tip If the < and > symbols are replaced with `<` and `>` when you paste, remove the pasted text, and then re-paste the schema by right clicking the design surface and selecting **Paste as HTML** from the shortcut menu.

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema id="dsCustomers" targetNamespace="http://www.tempuri.org/dsCustomers.xsd" xmln
s="http://www.tempuri.org/dsCustomers.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema" xm
lns:msdata="urn:schemas-microsoft-com:xml-msdata" elementFormDefault="qualified">
   <xs:element name="dsCustomers" msdata:IsDataSet="true">
      <xs:complexType>
         <xs:choice maxOccurs="unbounded">
            <xs:element name="Customers">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="CustID" type="xs:string" />
                     <xs:element name="CompName" type="xs:string" />
                     <xs:element name="ContName" type="xs:string" minOccurs="0" />
                     <xs:element name="ContTitle" type="xs:string" minOccurs="0" />
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
         </xs:choice>
      </xs:complexType>
         <xs:key name="Constraint1" msdata:PrimaryKey="true">
            <xs:selector xpath=".//Customers" />
            <xs:field xpath="CustID" />
         </xs:key>
      </xs:element>
   </xs:schema>
```

5. Save the schema and close it.
6. Check that you have a dataset class file based on the schema. In Solution Explorer, click the **Show All Files** button, and then expand the node for the new schema. You should see a class file (with extension of .jsl) underneath the schema.

Now that you have a schema and related dataset class file, you can add an instance of the dataset to your form.

To add the dataset to the form

1. Click the **Form1** tab at the top to return to the Windows Form Designer and your form.
2. From the **Data** tab of the Toolbox, drag the **DataSet** element onto the form.

   The **Add Dataset** dialog box is displayed.

3. Select **Typed Dataset** and then from the **Name** list, choose the dataset named **Walkthrough_Mappings.dsCustomers**. The dataset name is prefixed with your project name.

   An instance of the dataset is added to the form under the name dsCustomers1.

Creating a Data Adapter and Table Mapping

With the dataset defined, you can configure the other half of your data access story: the data adapter that will read data from the database and fill it into the dataset. The data adapter will contain the table mapping that matches the database columns to those in the dataset.

Adding the Data Adapter

The first step is to create the data adapter, which you can do using a wizard.

To add a data adapter

1. From the **Data** tab of the Toolbox, drag the **OleDbDataAdapter** element onto the form.

   The **Data Adapter Configuration Wizard** dialog box opens.

2. In the wizard, do the following:
   a. In the second page, create or choose a connection pointing to the SQL Server Northwind database.
   b. In the third page, specify that you want to use an SQL statement to access the database.
   c. In the fourth page, create the following SQL statement:

   ```
   SELECT CustomerID, CompanyName, ContactName, ContactTitle
   FROM Customers
   ```

   For assistance building the SQL statement, click **Query Builder** to launch the **Query Builder**.

   > **Note** In this walkthrough, you will populate the dataset with all the rows from the Customers table. In production applications, you typically optimize data access by creating a query that returns only the columns and rows you need. For an example, see Walkthrough: Displaying Data in a Windows Form Using a Parameterized Query with Visual J#.

When the wizard is complete, you will have a connection (**OleDbConnection1**) containing information about how to access your database. You will also have a data adapter (**OleDbDataAdapter1**) that contains a query defining what table and columns in the database you want to access.

Creating the Table Mapping

Now that you have a data adapter on the form, you can configure its table mappings.

To create the mappings

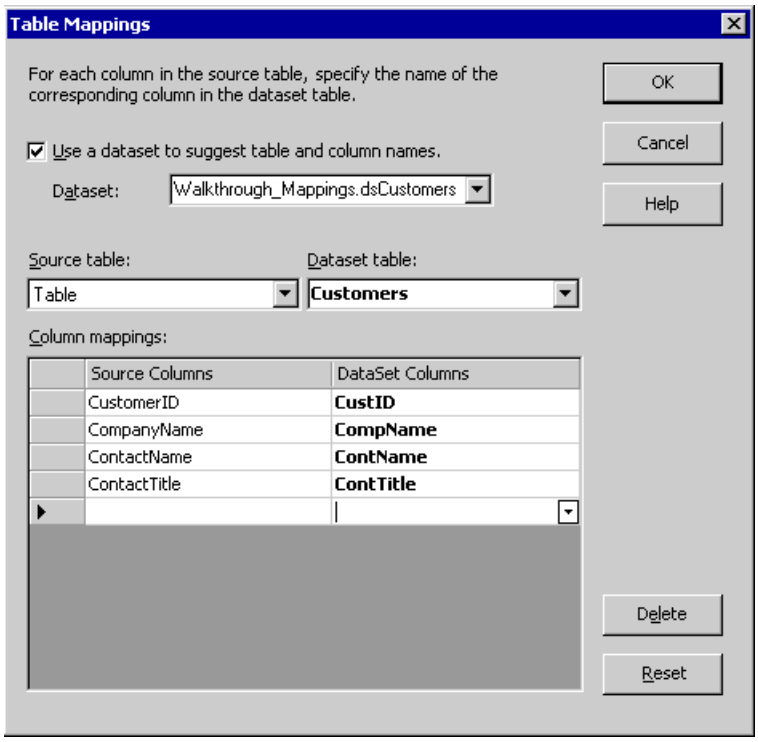
1. Select the data adapter at the bottom of the form.
2. In the Properties window, click the button in the **TableMappings** property.

   The **Table Mappings** dialog box is displayed.

3. Check **Use a dataset to suggest table and column names** and select **Walkthrough_Mappings.dsCustomers** from the drop-down list. (If you named your project something else, the prefix will be different in this list.)
4. Select **Customers** from the **Dataset table** list.
5. Change the names of columns in the **Dataset Columns** list individually by using the drop-down list that is available for each column. Create the following mappings:

Source Columns	Dataset Columns
CustomerID	CustID
CompanyName	CompName
ContactName	ContName
ContactTitle	ContTitle

When you have finished, the **Table Mappings** dialog box will look like this:



6. Click OK to close the **Table Mappings** dialog box.

Using Controls to Illustrate Mapped Names

In the final step of the walkthrough, you add a **DataGrid** control that illustrates how the mappings are used in the dataset. When you bind the grid to the dataset table, you will see that the dataset table contains the mapped names.

To add and bind a DataGrid control

1. Add a **DataGrid** from the **Windows Forms** tab of the **Toolbox**.
2. Select the **DataGrid** and open the **Properties** window.
3. Set the following data-binding properties:

   - Set the **DataSource** property to **dsCustomers1**.
   - Set the **DataMember** property to **Customers**.

   The column headings will match those in the schema.

To be able to run the form and see the mapped data in use, you will need to fill the dataset.

To fill the dataset

- Double-click the form (not the control) to create an event handler for the form's **Load** event and add the following line to it:

```
// Visual J#
oleDbDataAdapter1.Fill(dsCustomers1);
oleDbConnection1.Close();
```

Finally, you can test the form to see the mapping in action.

To test the form

1. Press F5 to run the form.
2. When the form is displayed, confirm that the data from the Northwind Customers table is displayed in the grid using the mapped column names.

Next Steps

This walkthrough has provided a brief introduction to table mapping in data adapters. Mapping can be used not just to change column names, but in more sophisticated ways as well, including:

- Map columns from one table to an entirely different column in the dataset.

- Use mapping to limit the number of columns created in a dataset. This is useful if your data adapter reads all the columns in a database table but you only need a few of them in a dataset table.
- Map an expression column in a dataset to a database column. This allows you to calculate values in the dataset and write them to the database.
- Raise errors under various circumstances, including mismatches.

For more information, see Table Mapping in Data Adapters.

See Also

Introduction to Data Adapters | Introduction to Datasets | Introduction to XML Schemas | Creating XML Schemas and Datasets | Creating an XML Schema with the XML Designer | Data Walkthroughs

Walkthrough: Reading XML Data into a Dataset with Visual J#

ADO.NET provides simple methods for working with XML data. In this walkthrough you will create a Windows application that will load XML data into a dataset. The dataset will then be displayed in a **DataGrid** control. Finally, an XML schema based on the contents of the XML file will be displayed in a text box.

This walkthrough consists of five main steps:

1. Create a new Visual J# project.
2. Create an XML file to be read into the dataset.
3. Create the user interface.
4. Add code to create the dataset, read the XML file, and display it in a **DataGrid** control.
5. Add code to display the XML schema based on the XML file in a **TextBox** control.

Create a New Project

In this step, you will create a Visual J# project that will contain this walkthrough.

To create a new project

1. From the **File** menu, point to **New**, and then click **Project** to display the **New Project** dialog box.
2. Select **Visual J# Projects** in the **Project Types** pane, and then select **Windows Application**.
3. Name the project **ReadingXML**, then click **OK**.

   Visual Studio will add the **ReadingXML** project to Solution Explorer and display a new Windows Form in the designer.

Generate the XML File to Be Read into the Dataset

Because this walkthrough focuses on reading XML data into a dataset, the contents of an XML file are provided. This XML file must be located in a folder that will be accessible from the machine that you are using to create this application.

To create the XML file that will be read into the dataset

1. From the **File** menu, point to **New**, and then click **File**.

   The **New File** dialog box appears.

2. Select **XML File** and click **Open**.

   The XML file loads into the designer and is ready for edit.

3. Paste the following code into the editor below the XML declaration:

   Tip Right-click the editor and select **Paste as HTML**.

```
<Authors_Table>
  <authors>
    <au_id>172-32-1176</au_id>
    <au_lname>White</au_lname>
    <au_fname>Johnson</au_fname>
    <phone>408 496-7223</phone>
    <address>10932 Bigge Rd.</address>
    <city>Menlo Park</city>
    <state>CA</state>
    <zip>94025</zip>
    <contract>true</contract>
  </authors>
  <authors>
    <au_id>213-46-8915</au_id>
    <au_lname>Green</au_lname>
    <au_fname>Margie</au_fname>
    <phone>415 986-7020</phone>
    <address>309 63rd St. #411</address>
```

```
      <city>Oakland</city>
      <state>CA</state>
      <zip>94618</zip>
      <contract>true</contract>
    </authors>
    <authors>
      <au_id>238-95-7766</au_id>
      <au_lname>Carson</au_lname>
      <au_fname>Cheryl</au_fname>
      <phone>415 548-7723</phone>
      <address>589 Darwin Ln.</address>
      <city>Berkeley</city>
      <state>CA</state>
      <zip>94705</zip>
      <contract>true</contract>
    </authors>
    <authors>
      <au_id>267-41-2394</au_id>
      <au_lname>Hunter</au_lname>
      <au_fname>Anne</au_fname>
      <phone>408 286-2428</phone>
      <address>22 Cleveland Av. #14</address>
      <city>San Jose</city>
      <state>CA</state>
      <zip>95128</zip>
      <contract>true</contract>
    </authors>
    <authors>
      <au_id>274-80-9391</au_id>
      <au_lname>Straight</au_lname>
      <au_fname>Dean</au_fname>
      <phone>415 834-2919</phone>
      <address>5420 College Av.</address>
      <city>Oakland</city>
      <state>CA</state>
      <zip>94609</zip>
      <contract>true</contract>
    </authors>
  </Authors_Table>
```

4. From the **File** menu, point to **Save XMLFile1 As**.

   The **Save File As** dialog box appears.

5. Name the file **authors.xml** and click **Save**.

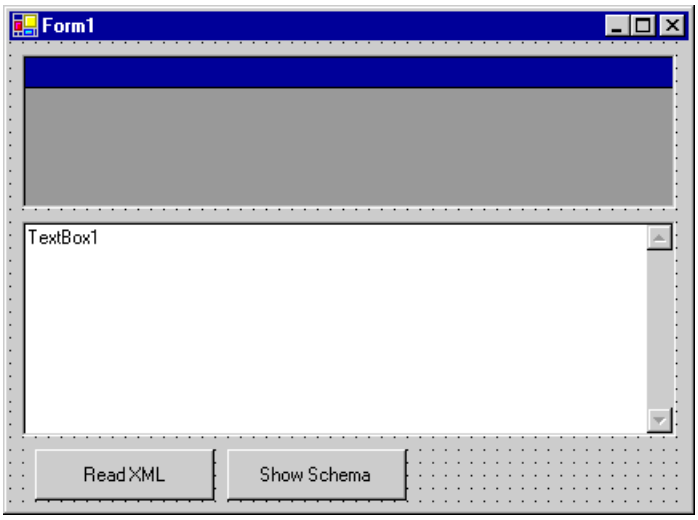
   Note Be sure to remember the complete path, it will be needed later in this walkthrough.

Create the User Interface

The user interface for this application will consist of the following:

- A **DataGrid** control that will display the contents of the XML file as data.
- A **TextBox** control that will display the XML schema for the XML file.
- Two **Button** controls, as follows:
  - One button reads the XML file into the dataset and displays it in the **DataGrid** control.
  - A second button extracts the schema from the dataset, and through a StringWriter displays it in the **TextBox** control.

Initial layout for this application



To add controls to the form

1. From the **Windows Forms** tab of the Toolbox, drag the following controls onto the form:
   - 1 DataGrid control
   - 1 TextBox control
   - 2 Button controls
2. Set the following properties:

Control	Property	Setting
TextBox1	**Multiline**	**true**
	ScrollBars	**Vertical**
Button1	**Name**	**btnReadXML**
	Text	**Read XML**
Button2	**Name**	**btnShowSchema**
	Text	**Show Schema**

Create the Dataset that will Receive the XML Data

You will programmatically create a new dataset named **authors**. It will be declared at form level in order to be accessible from the **button_click** event handlers. For more information about datasets, see Introduction to Datasets.

To add the code to create a new dataset that will receive the XML data

1. With **Form1.jsl** selected in Solution Explorer, click the **View Code** button.
2. In the declarations area, declare a new dataset named authors:

```
// Visual J#
DataSet dsAuthors = new DataSet("authors");
```

Create the Event Handler to Read the XML into the Dataset

The **Read XML** button reads the XML file into the dataset and sets properties on the **DataGrid** that bind it to the dataset.

To add code to the btnReadXML_Click event handler

1. In Solution Explorer, select **Form1** and click the **View Designer** button.
2. Double-click the **Read XML** button.

   The Code Editor opens at the **btnReadXML_Click** event handler.

3. Type the following code into the **btnReadXML_Click** event handler:

```
// Visual J#
   System.String filePath = "Complete path where you saved the XML file";
   dsAuthors.ReadXml(filePath);
```

```
    dataGrid1.set_DataSource(dsAuthors);
    dataGrid1.set_DataMember("authors");
    dataGrid1.set_CaptionText(dataGrid1.get_DataMember());
```

4. In the **btnReadXML_Click** event handler code, edit the `filepath =` entry to the correct path.

Create the Event Handler to Display the Schema in the Textbox

The Show Schema button creates a **StringWriter** object that is filled with the schema and displayed in the **TextBox**.

To add code to the btnShowSchema_Click event handler

1. In Solution Explorer, select **Form1.jsl** and click the **View Designer** button.
2. Double-click the **Show Schema** button.

   The Code Editor opens at the **btnShowSchema_Click** event handler.

3. Type the following code into the **btnShowSchema_Click** event handler.

```
// Visual J#
   System.IO.StringWriter swXML = new System.IO.StringWriter();
   dsAuthors.WriteXmlSchema(swXML);
   textBox1.set_Text(swXML.ToString());
```

Run the Application

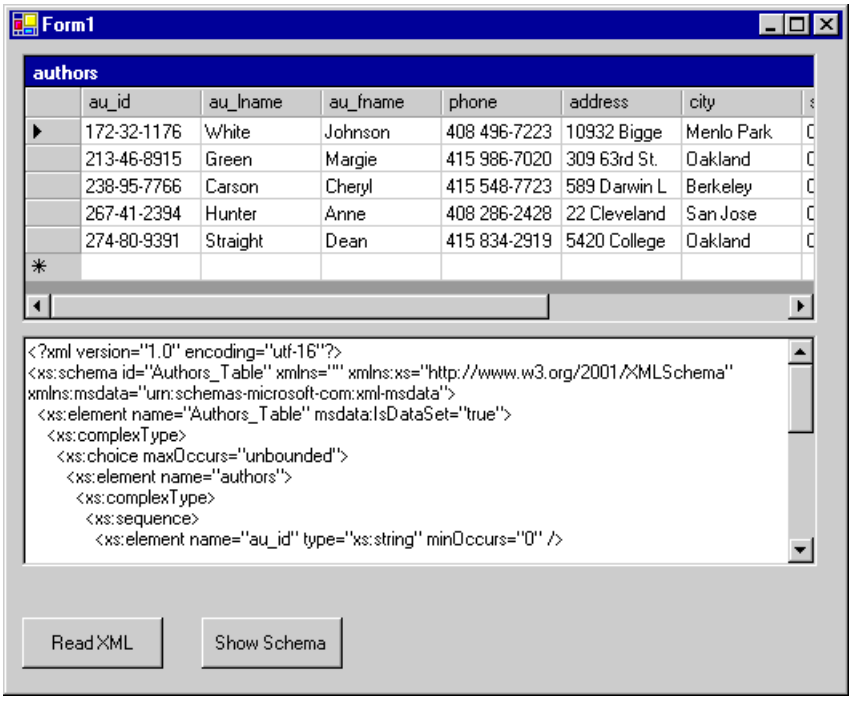
1. Press F5 to run the application.
2. Click the **Read XML** button.

   The data grid displays the contents of the XML file.

3. Click the **Show Schema** button.

   The text box displays the XML schema for the XML file.

Result of running the application



Next Steps

This walkthrough shows the basics of reading an XML file into a dataset, as well as creating a schema based on the contents of the XML file. Here are some tasks that might come next:

- Edit the data in the dataset and write it back out as XML. For more information, see DataSet.WriteXml Method.

- Edit the data in the dataset and write it out to a database. For more information, see Updating a Data Source with a Dataset.

See Also

Introduction to Datasets | DataSet.ReadXml Method | DataSet.WriteXmlSchema Method | Debugging Managed Code | Deploying Applications | XML in Visual Studio | Data Walkthroughs

Walkthrough: Switching Windows Forms Menu Structures Based on Application State with Visual J#

Your application can present different menus for different contexts (or different application states). It is possible to have many MainMenu objects, each of which presents different menu choices to the user. By having a number of **MainMenu** objects to present the correct menu structure to the user, you can handle the varying states of your application as users interact with it.

In this procedure, you will create a menu structure for when the application first opens and there is no file or data for the user to interact with. Thus, the application will have only a traditional File menu with New, Open, and Exit commands. When the user selects either the New or Open menu item, it triggers a change in application state (for demonstration purposes in the following example, a message box is shown as a visual aid; no actual files are opened or created). At this point, a second menu structure is shown, with additional menu items (Close and Save) that target an application that has data or a file loaded.

Creating a Menu at Design Time

In the following steps, you will design a Windows application, in the Windows Forms Designer, that switches menu structures.

To create a menu at design time

1. Create a new Windows Application project.
2. Drag a **MainMenu** component from the Toolbox to the form.
3. In the Menu Designer, create a top-level menu item with the **Text** property set to `&File`, and three submenu items with the **Text** properties set to `&New`, `&Open`, and `E&xit`, in that order. For details, see Adding Menus and Menu Items to Windows Forms.
4. In the Windows Forms Designer, click the form to give it focus. In the Properties window, ensure that the **Menu** property is set to the menu you have just created (**mainMenu1**, unless you have given it a unique name).

To hook up the events

1. Create a multicast event handler called **MenuSelect** for the **New** and **Open** menu items. You need to add references to the event handler in the **form** initialization function for Visual J#, as follows:

```
// Visual J#
this.menuItem2.add_Click( new System.EventHandler(this.MenuSelect) );
this.menuItem3.add_Click( new System.EventHandler(this.MenuSelect) );
```

   This is a multicast event handler because it will handle the **Click** events for both menu items. For details, see Connecting Multiple Events to a Single Event Handler in Windows Forms.

2. Enter the following code in the **MenuSelect** event handler:

```
// Visual J#
private void MenuSelect(Object sender, System.EventArgs e)
{
   MessageBox.Show("A file has been opened.", "Instead of a new file, here's a message bo
x.");
   LoadSecondMenu();
}
```

   Note `LoadSecondMenu` will be a procedure you write to switch the menu when either of the menu items (**New** or **Open**) is clicked.

3. In the Code Editor, create a method, similar to the following, to set the form's menu to the second **MainMenu** component you created:

```
// Visual J#
private void LoadSecondMenu()
{
   this.set_Menu ( mainMenu2);
}
```

Note `MainMenu2` refers to a **MainMenu** component you add to the form later on.

To copy the menus for the second set

1. In Solution Explorer, right-click the form name and choose **View Designer** on the shortcut menu.
2. In the component tray, click the icon for the **MainMenu** component, so that it has focus in the form.
3. In the Menu Designer, click the top-level menu item you created (**File**). Right-click and choose **Copy**.

   Note This makes a copy of the top-level menu item and all of its submenu items.

4. Drag another **MainMenu** component from the Toolbox to the form.
5. In the Menu Designer, right-click the "Type Here" area and choose **Paste**.

   The menu items you previously selected from the first **MainMenu** component are pasted into the second.

6. In the Menu Designer, right-click the area to the left of the **Exit** menu item and choose **Insert New**. Do this again, so that you have inserted two new menu items.
7. Set the **Text** property of these two new menu items to `&Close` and `&Save`, in that order, either in the Properties window or by clicking the item and typing in the space provided.

To test the application

- Press **F5** to run the application. Debug it if necessary. For details about debugging, see Debugging Basics.

  The form has a menu that contains **File**, **New**, **Open**, and **Exit** menu items. Clicking **New** or **Open** raises an event, which is handled by the **MenuSelect** event handler. This method shows a message box and changes the application state. The application state change is indicated by the addition of two menu items, **Close** and **Save**.

Instead of using the above procedure, you can create the entire application in code. While this forgoes the ease of using the Windows Forms Designer, some developers may find this method more appealing.

Creating a Menu Programmatically

In the following steps, you will design a Windows application (with identical results to the one above) exclusively through code that switches menu structures.

To create a menu programmatically

1. Create a new Windows Application project.
2. In a method, create an instance of the **MainMenu** component with **MenuItem** objects corresponding to the application state.

   The following example has one **MainMenu** object for when the user first opens the application; it has only a traditional **File** menu with **New**, **Open**, and **Exit** commands. Additionally, the code below uses an overloaded Add method that creates menu items and associated event handlers with them. Wiring event handlers dynamically is an easy way to signal to the application that clicking either the **New** or **Open** menu command should trigger a change in the application state.

```
// Visual J#
// Create a MainMenu object and a MenuItem object.
MainMenu mmAppStart;
MenuItem miFile;
public void AppStartMenu()
{
   // Create an instance of the MainMenu object.
   mmAppStart = new MainMenu();

   // Create a top-level menu item and two menu items. Use this
   // overloaded constructor that takes an event handler
   // (MenuSelect) so that later, you can cause the menu selection
   // to change the application state.
   miFile = new MenuItem("&File", new System.EventHandler(MenuSelect));
   miFile.get_MenuItems().Add("&New", new System.EventHandler(MenuSelect));
```

```
    miFile.get_MenuItems().Add("&Open", new System.EventHandler(MenuSelect));
    miFile.get_MenuItems().Add("&Exit");

    // Add the top-level menu item to the MainMenu component
    // and set the MainMenu component to be the form's menu.
    mmAppStart.get_MenuItems().Add(miFile);

    // Set the form's menu to the menu you have just created.
    this.set_Menu ( mmAppStart);
}
```

To create the second set of menus

1. In a second method, create a second instance of a **MainMenu** component with **MenuItems** that correspond to a second application state.

   For example, after the user has opened a file, you can expose commands to close it and save it. You can copy the original menu structure to use as a base for this menu structure, and to preserve the **File**, **New**, **Open**, and **Exit** menu items. For details, see Copying Items on Windows Forms Menus.

```
// Visual J#
// Create the second MainMenu object
private MainMenu mmFileLoadedMenu;
private void FileLoadedMenu()
{
    // Clone first menu
    mmFileLoadedMenu = mmAppStart.CloneMenu();

    // Add Close and Save menu items to the new menu's File menu items,
    // using the Add method to specify their order within the collection.
    MenuItem miFile = mmFileLoadedMenu.get_MenuItems().get_Item(0);
    miFile.get_MenuItems().Add(2, new MenuItem("&Close"));
    miFile.get_MenuItems().Add(3, new MenuItem("&Save"));

    // Assign the newly-created MainMenu object to the form.
    this.set_Menu(mmFileLoadedMenu);
}
```

2. Add a line of code to the `Form1` constructor, after the `InitializeComponent` method call, to call the `AppStartMenu` method created previously:

```
// Visual J#
AppStartMenu();
```

3. Create an event handler within the class to switch the form's menu property to `FileLoadedMenu`.

   Instead of loading a file, the following code opens a message box to indicate a change in application state. For details, see MessageBox Class. The code then calls the `AppStartMenu` method that was written above, which creates the second **MainMenu** object and sets it to be the form's menu.

```
// Visual J#
protected void MenuSelect(Object sender, System.EventArgs e)
{
    MessageBox.Show("A file has been opened.", "Instead of a new file, here's a message bo
x.");
    FileLoadedMenu();
}
```

To test the application

- Press **F5** to run the application. Debug it if necessary. For details about debugging, see Debugging Basics.

  The form has a menu that contains **File**, **New**, **Open**, and **Exit** menu items. Clicking **New** or **Open** raises an event, which is handled by the **MenuSelect** event handler. This method shows a **MessageBox** and changes the application state. The application state change is indicated by the addition of two menu items, **Close** and **Save**.

See Also

Introduction to the Windows Forms MainMenu Component | Menus in Windows Forms | Windows Forms Walkthroughs

Walkthrough: Simple Data Access in a Windows Form with Visual J#

One of the most common scenarios in application development is to display data on a form. This walkthrough illustrates a simple Windows Form that displays data from a single table in a data grid. The grid is editable, and you will be able to make changes to data and update the database. Although the result is not complex, the walkthrough illustrates many of the basic procedures you will use when accessing data with forms.

To complete this walkthrough, you will need:

- Access to a server with the Pubs SQL Server sample database.

The walkthrough is split into a number of smaller pieces:

- Creating the Windows Form.
- Creating and configuring the dataset you will bind the form against. This includes creating a query that populates the dataset from the database.
- Adding the **DataGrid** control to the form and binding it to data.
- Adding code to fill the dataset.
- Adding code that sends dataset changes back to the database.

When you have finished the walkthrough, you will have a form that looks similar to the following.

Creating the Project and Form

The first step is to create a Windows Form.

To create the project and form

1. From the **File** menu, point to **New**, and then choose **Project**.
2. In the **Project Types** pane, choose **Visual J# Projects**, and then in the **Templates** pane, choose **Windows Application**.
3. If you already have a solution open, select **Close Solution**.

   Note In production projects, you will very often have multiple projects in the same solution. In this walkthrough, however, you will close any open solution and create a new one along with the project so that there is no interference between what you will do here and any existing forms, datasets, and so on.

4. Assign a name to the project that will be unique and will conform to the naming conventions you use. For example, you might name this project **Walkthrough_Simple1**.
5. When you have assigned a name and specified a new solution, click **OK**.

   Visual Studio creates a new project and displays a new form in the Windows Form Designer.

Creating and Configuring a Dataset

As with most data-access scenarios in Windows Forms applications, you'll be working with a dataset. A dataset is a container — a cache — that holds the records you are interested in working with.

> **Note** Using a dataset is only one option for data access, and is not the optimal choice in some scenarios. Nonetheless, datasets are usually the right choice in Windows Forms applications, and you will use one in this walkthrough. For more information, see Recommendations for Data Access Strategies.

In this walkthrough, you will add a dataset to the form. However, because the dataset does not already exist, you will not manually add it to the form. Instead, you will perform the following set of steps:

- Create a data adapter using a wizard. The adapter contains SQL statements used to read and write database information. The wizard helps you define the SQL statements you need. If necessary, the wizard also creates a connection to the database.
- Generate the dataset schema. In this process, you will have Visual Studio create a new typed dataset class based on the tables and columns you are accessing. When you generate the dataset class, you will also add an instance of it to the form.

It is important that you follow all procedures in this section. Otherwise your form will not have the dataset that you will be using in subsequent parts of the walkthrough.

For more information on data adapters, see Introduction to Data Adapters. For more information about datasets, see Introduction to Datasets.

Configuring a Data Connection and Data Adapter

To begin, you create a data adapter that contains the SQL statement used to populate the dataset later. As part of this process, you define a connection to access a database. You configure the data adapter using a wizard, which makes it easy to create the SQL statements you need for data access.

> **Note** When the wizard is done, you must continue to the next section in order to generate a dataset and complete the data access portion of your form.

To create the data connection and data adapter

1. From the **Data** tab of the **Toolbox**, drag an **OleDbDataAdapter** object onto the form.

   > **Note** You could also use the **SqlDataAdapter**, which is optimized for working with SQL Server 7.0 or later. In this walkthrough, you use the **OleDbDataAdapter** because it is more generic, providing ADO.NET access to any OLE DB-compatible data source.

   The **Data Adapter Configuration Wizard** starts, which will help you create both the connection and the adapter.

2. In the wizard, do the following:
   a. In the second page, create or choose a connection pointing to the SQL Server Pubs database. (To get to the second page, click **Next** in the first page.)
   b. In the third page, specify that you want to use an SQL statement to access the database.
   c. In the fourth page, create the following SQL statement:

   ```
   SELECT au_id, au_lname, au_fname, city, state, phone, contract
   FROM authors
   ```

   For assistance building the SQL statement, click **Query Builder** to launch the **Query Builder**.

   > **Note** In this walkthrough, you will populate the dataset with all the rows from the authors table. In production applications, you typically optimize data access by creating a query that returns only the columns and rows you need. For an example, see Walkthrough: Displaying Data in a Windows Form Using a Parameterized Query with Visual J#.

   The wizard creates a connection (**OleDbConnection1**) containing information about how to access your database. You will also have a data adapter (**OleDbDataAdapter1**) that contains a query defining what table and columns in the database you want to access.

3. After the wizard is complete, generate the dataset based on the SQL query that you created during this procedure. For more information, see the next section.

Creating the Dataset

After you have established the means to connect to the database and specified the information you want (via the SQL command in the data adapter), you can have Visual Studio create a dataset. Visual Studio can generate the dataset automatically based on the query you specified for the data adapter. The dataset is an instance of the **DataSet** class based on a corresponding XML Schema (.xsd file) that describes the class's elements (table, columns, and constraints). For more information about the relationship between datasets and schemas, see Introduction to Data Access with ADO.NET.

To generate a dataset

1. From the **Data** menu, choose **Generate DataSet**.

   > **Tip** If you do not see the **Data** menu, click in the form; the form must have focus for the menu to appear.

   The **Generate Dataset** dialog box is displayed.

2. Select the **New** option and name the dataset **dsAuthors**.

   In the list under **Choose which table(s) to add to the dataset**, the authors table should be selected.

3. Make sure **Add this dataset to the designer** is checked, and then click **OK**.

   Visual Studio generates a typed dataset class (**dsAuthors**) and a schema that defines the dataset. You will see the new schema (**dsAuthors.xsd**) in **Solution Explorer**.

   > **Tip** In Solution Explorer, click **Show All Files** to see the schema file's dependent .jsl file, which contains the code the defines your new dataset class.

   Finally, Visual Studio adds an instance of the new dataset class (**dsAuthors1**) to the form.

At this point you have set up everything you need in order to get information out of the database and into a dataset. You are ready to create a form that will display the data.

Adding a DataGrid Control to Display the Data

In this walkthrough, you will add a single control — a **DataGrid** — that can display all the records from the dataset at the same time. An alternative would be to use individual controls such as text boxes to display one record at a time. That strategy then requires you to add navigation to the form. For simplicity, therefore, you will use a data grid.

> **Note** For an example of how to use individual text boxes to display records from a dataset, see Walkthrough: Displaying Data in a Windows Form Using a Parameterized Query with Visual J#.

The data grid must be bound to the dataset in order to display the data.

To add a bound DataGrid control to the form

1. If you have not already done so, switch to the form designer by clicking the tab at the top of the current window.
2. From the **Windows Forms** tab of the Toolbox, drag a **DataGrid** control onto the form.
3. Press **F4** to display the Properties window.
4. In the **DataSource** property, select **DsAuthors1** as the data source. (Do not choose **DsAuthors1.authors**.)
5. In the **DataMember** property, select **authors**.

   Setting these two properties binds the authors data table in the **DsAuthors1** dataset to the grid.

6. Resize the grid so you can see all the columns. Change its height so you will be able to see several author records.

Populating the DataGrid Control

Although the data grid is bound to the dataset you created, the dataset itself is not automatically filled with data. Instead, you must fill the dataset yourself by calling a data adapter method. For more information about filling datasets, see Introduction to Datasets.

To populate the DataGrid control

1. From the **Windows Forms** tab of the Toolbox, drag a **Button** control onto the form.
2. Name the button **btnLoad** and change the caption by setting its **Text** property to **Load**.
3. Double-click the button to create an event-handling method for its **Click** event.

4. In the method, clear the dataset you created, and then call the data adapter's **Fill** method, passing it the dataset you want to fill.

   The following example shows what the complete method will look like:

```
// Visual J#
private void btnLoad_Click(Object sender, System.EventArgs e)
{
   dsAuthors1.Clear();
   oleDbDataAdapter1.Fill(dsAuthors1);
   oleDbConnection1.Close();
}
```

Updating the Database

When users make a change in the grid, the control automatically saves the updated record in the dataset. In Windows Forms, the data-binding architecture writes the values of data-bound controls to the data rows they are bound to.

> **Note** In Web Form pages, data binding works somewhat differently. For a step-by-step example of data binding in Web Forms pages, see Walkthrough: Displaying Data in a Web Forms Page with Visual J# and Walkthrough: Updating Data Using a Database Update Query in Web Forms with Visual J#.

However, when you work with a dataset, updates require two stages. After the data is in the dataset, you still have to send it from the dataset to the database. The data adapter can do this with its **Update** method, which examines every record in the specified data table in the dataset and, if a record has changed, sends the appropriate Update, Insert, or Delete command to the database. For more information, see Introduction to Dataset Updates.

In this walkthrough, you will add a button to the form that users can press when they want to send their updates to the database.

To update the database

1. From the **Windows Forms** tab of the Toolbox, drag a **Button** control onto the form.
2. Name the button **btnUpdate** and change the caption by setting its **Text** property to **Save Changes in Database**.
3. Double-click the button to create an event-handling method for its **Click** event.
4. In the method, call the data adapter's **Update** method, passing it the dataset containing the updates you want to send to the database. Use the **MessageBox** object to display confirmation text.

   The following example shows what the complete method will look like:

```
//Visual J#
private void btnUpdate_Click(Object sender, System.EventArgs e)
{
   oleDbDataAdapter1.Update(dsAuthors1);
   oleDbConnection1.Close();
   MessageBox.Show("Database updated!");
}
```

Testing

You can now test the form to make sure it displays authors data in the grid and that users can make updates.

To test the form

1. Press **F5** to run the form.
2. When the form is displayed, click the **Load** button.

   A list of authors is displayed in the grid.

3. Make a change in a record in the grid.

   When you move to another record in the grid, the change is preserved. Make a mental note of your change.

4. Click the **Load** button again.

This reloads the dataset from the database and refreshes the grid. Notice that the change you made in Step 3 was not preserved — because you did not save changes from the dataset to the database.

5. Make a change in a record in the grid again.
6. Click the **Save Changes in Database** button.

   You will see the message box displayed, but there is no change in the grid.

7. Click the **Load** button once more to reload data from the database.

   This time, the change you made in Step 5 is preserved, because the data was saved in the database.

Next Steps

This walkthrough has illustrated the basic steps involved in displaying data in a form. Some enhancements you could make to the form in this walkthrough include:

- Format the grid by changing its color, font, and so on.
- Make the grid display data without being explicitly loaded by the user. To do this, you add a **Fill** method to the form's constructor and remove the **Load** button.
- Display only selected information in the grid. In many instances, you will base the display on information that the user supplies at run time (for example, you might display authors only from a particular city). To do this, you create a parameterized query. For more information, see Walkthrough: Displaying Data in a Windows Form Using a Parameterized Query with Visual J#.
- Separate data access from the user interface. In this walkthrough, you have created a form that accesses the data more-or-less directly (via the dataset). A more flexible and maintainable design is to create a data-access component that handles data access. The form (that is, the user interface) could then interact with the component as needed. The same component could be used by multiple forms (and by other components), which eliminates the overhead and redundancy of redesigning data access for every form you create. For more information on creating component-based data access, see Walkthrough: Creating a Distributed Application with Visual J#.

See Also

Windows Forms Walkthroughs

Walkthrough: Validating User Input in a Web Forms Page with Visual J#

As with any input form, Web Forms pages require you to make sure that users are entering information that is valid, is in the right format, and so on. To make this common task easier, you can use validation controls created especially for Web Forms pages. Validation controls allow you to check for required entries, data types, ranges, patterns, and specific values by simply setting properties. The controls perform checking automatically and include a variety of ways for displaying error information to users. For an overview, see Web Forms Validation.

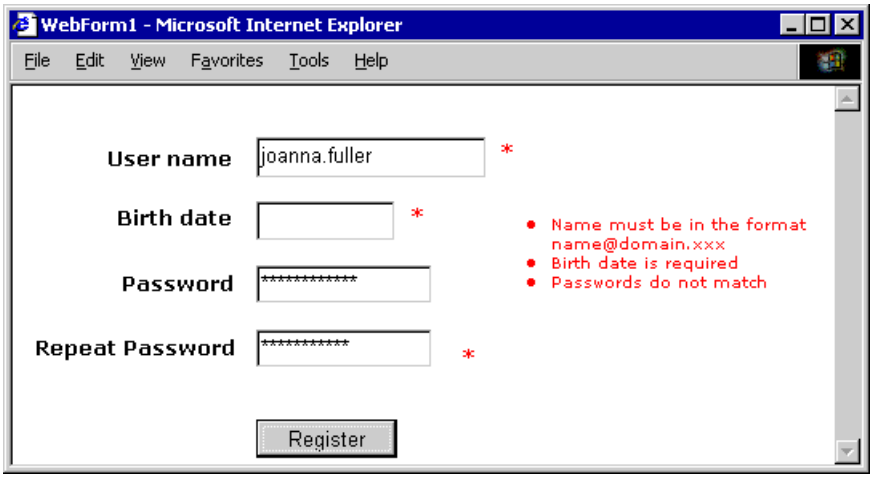
This walkthrough illustrates a simple Web Forms page that you might create as a simple registration page. The page includes text boxes to allow users to enter a login name, a birth date, a password, and a repeat of the password. You will add validation controls to the page to be sure that users enter values into all the boxes and that the values are formatted correctly.

To complete this walkthrough, you will need sufficient permissions to be able to create ASP.NET Web application projects on the computer where your Web server is.

The walkthrough is split into a number of smaller pieces:

- Creating the basic form by creating a project and Web Forms page and adding the input controls and labels.
- Adding the validation controls. You will validate for required entries, that the user has entered a valid login name and date, and that the passwords match.
- Testing.

When you have finished the walkthrough, you will have a form that looks similar to the following.



Creating the Basic Form

You can start by creating a form with input controls on it.

Creating the Form

The first step is to create a form that you will later add validation controls to. Start by creating a new Web application project and Web Forms page.

To create the project and form

1. On the **File** menu, point to **New**, then click **Project**.
2. In the **New Project** dialog box, do the following:
   a. In the **Project Types** pane, choose **Visual J# Projects**.
   b. In the **Templates** pane, choose **ASP.NET Web Application**.
   c. In the **Location** box, enter the complete URL for your application, including http://, the name of the server, and the name of your project. The Web server must have IIS version 5 or later and the .NET Framework installed on it. If you have IIS on your computer, you can specify `http://localhost` for the server.

      When you click **OK**, a new Web Forms project is created at the root of the Web server you specified. In addition, a new Web Forms page called WebForm1.aspx is displayed in the Web Forms Designer in Design view.

Tip If you have trouble creating a Web application project, see Web Access Failed Dialog Box.

3. If the page is not already in grid mode, use the Properties window to set the document object's **pageLayout** property to **GridLayout**.

   This step is optional, but makes it easier to lay out the form. If you prefer to work in flow layout, you can, but you might have to make minor adjustments to where the controls appear on the page. The page will work the same way in either case.

Adding Input Controls

For this walkthrough, which assumes that you are creating a simple registration page, you will add controls to allow users to enter a name (an e-mail address), a birth date, and a password. Users enter the password twice to confirm it.

To add the controls

1. From the **Web Forms** tab of the Toolbox, add the following controls to the form, setting their **ID** property as indicated:

Control	Purpose	ID
TextBox	Enter user name	**txtName**
TextBox	Enter birth date	**txtBirthdate**
TextBox	Enter password	**txtPassword**
TextBox	Repeat password to confirm it	**txtPasswordAgain**
Button	Submit form	**btnRegister** Set the button's **Text** property to Register .

2. Set the **TextMode** property of the two password boxes to **Password** so that they display stars (*) as users type in them.
3. Add labels in front of the text boxes to indicate their function. If the Web Forms page is in grid mode, add **Label** controls in front of each text box and set their **Text** property. If the page is in flow mode, simply type static text in front of each box.
4. Adjust the font, size, and layout of the controls to your liking. For details about layout, see Positioning HTML Elements in Design view.

Adding Validation Controls

Now that you have created the input form, you can add validation. You do so by adding validation controls to the form, one control for each validation test you want to perform. (In all, you will add seven validation controls to cover all the tests you should perform plus a control to display errors.) The validation tests you will perform are these:

- Required fields. You want to be sure that users enter values into all four boxes.
- Name format. For this walkthrough, you will check that the user's login name follows the pattern of typical e-mail names: *name@domain.xxx*. In this walkthrough, you will not validate that the e-mail name the user provides belongs to a valid e-mail account.
- Date format. You will validate that the user's entry for birth date is a valid date. (In this walkthrough, you will not validate that the date falls within a specific range.)
- Password match. You will validate that the passwords that the user supplies in the two password boxes match.

Checking For Required Fields

You want to be sure that users enter values into all four text boxes — that is, all four text boxes are required fields.

To check for required fields

1. From the **Web Forms** tab of the Toolbox, drag a **RequiredFieldValidator** control onto the form, next to the **txtName** text box.
2. Select the validation control and in the Properties window, set the following properties.

Property	Setting
ControlToValidate	`txtName` This indicates that the validator will check the contents of the name field.

Text	`*` (star) This is the text that will be displayed where the validator control is now. (It is common to put a red star next to an input box containing errors.)
ErrorMessage	`Email name is required.` This text will be displayed in a summary on the page.

3. If you like, set the font, color, and formatting properties for the validation control. These properties affect how errors are displayed.
4. Repeat steps 1 through 3 for the other three text boxes, creating three more **RequiredFieldValidation** controls with the following settings.

Control	Settings
RequiredFieldValidator2	• **ControlToValidate**: Set to `txtBirthdate` • **Text**: `*` • **ErrorMessage**: `You must enter a birth date.`
RequiredFieldValidator3	• **ControlToValidate**: Set to `txtPassword` • **Text**: `*` • **ErrorMessage**: `You must enter a password.`
RequiredFieldValidator4	• **ControlToValidate**: Set to `txtPasswordAgain` • **Text**: `*` • **ErrorMessage**: `Re-enter the password to confirm it.`

Checking Name Format

For the walkthrough, you are assuming that the user should enter an e-mail address as the registration name. Although you cannot check that the name represents a valid e-mail account, you can check that the name conforms to typical e-mail address format (such as *name@domain.com*). This catches entry errors that users often make, such as forgetting to enter the ".com" portion of their e-mail address.

To check for patterns such as this, you use a validator control that incorporates a regular expression. If you know regular expression syntax, you can create your own pattern-matching expressions. However, you can also choose from a list of predefined regular expressions, including those to check for e-mail addresses, phone numbers, and postal codes.

When you add this validation control, you will have two controls for the name box; the contents of the box must pass both checks to be considered valid. When you have multiple validators for a single control, you generally want them to display their text (typically, a star) in the roughly the same location, no matter which validation check fails. You can specify this by setting the validator control's Display property. For more information, see Validation Error Message Layout for ASP.NET Server Controls.

To check name format

1. From the **Web Forms** tab of the Toolbox, drag a **RegularExpressionValidator** control next to the **txtName** box.
2. Select the validation control and in the Properties window, set the following properties.

Property	Setting
ControlToValidate	`txtName`
Text	`*` (star)
ErrorMessage	`Name must be in the format name@domain.xxx.`
ValidationExpression	From the list, select **Internet E-mail Address**. Note that this sets the property val ue to a regular expression.

Checking Date Format

You want to be sure that users enter a valid date for the birth date field. You can do this with a validation control that performs two functions at once: it checks that the user has entered a date (by data type) and it compares that the date is later than a specified minimum date.

To check date format

1. From the **Web Forms** tab of the Toolbox, drag a **CompareValidator** control next to the **txtBirthdate** box.
2. Select the validation control and in the Properties window, set the following properties.

Property	Setting
ControlToValidate	`txtBirthdate`
Text	* (star)
ErrorMessage	`Birth date is not a valid date.`
ValueToCompare	`01/01/1900` (Or any other minimum date.)
Operator	`Greater Than`
Type	`Date`
Display	**Dynamic**

Checking for Password Match

As is done often, your registration form asks users to type their password twice to confirm that that is what they want to enter. You can check this using the same validation control you used for the date — a comparison control — but instead of checking against a specific value, you will compare the value of one field — the second password box — against what they typed into the first password box.

To check for password match

1. From the **Web Forms** tab of the Toolbox, drag a **CompareValidator** control next to the **txtPasswordAgain** box.
2. Select the validation control and in the Properties window, set the following properties.

Property	Setting
ControlToValidate	`txtPasswordAgain`
Text	* (star)
ErrorMessage	`The passwords do not match.`
ControlToCompare	`txtPassword`
Operator	`Equal`
Display	**Dynamic**

Displaying Validation Errors

One step remains. You have configured the validation controls so that they display a red star next to an input box containing an error. However, this might not give users enough information. For example, they might realize that they have entered a bad registration name, but might not be able to figure out what is wrong with it.

To display details about validation errors, you can use a summary control. The control displays the value of the **ErrorMessage** property of any validation control that has detected an error. If there is more than one error, the summary control can display them all, either as bulleted items, in a simple list, or as a paragraph of text.

To display validation errors

1. From the **Web Forms** tab of the Toolbox, drag a **ValidationSummary** control onto the form. Position the control someplace where it has room to display several lines of text.
2. If you want to display errors in a format other than bulleted items, select a different display option for the **DisplayMode** property.
3. Set the **Font** and **ForeColor** properties to format the error message text.
4. You are done adding validation controls. Now you can test to see how they work.

Testing Validation

You want to test the Web Forms page with different values to see the effect of the validation controls.

To test the validation controls

1. In Solution Explorer, right-click the name of your form and choose **View in Browser**.
2. When the page is displayed, enter various values into the boxes. To check the birth date, enter a date that is invalid for the locale of your Web server.

Each time you make an error, a star should appear next to the box containing the error.

3. Click **Register**.

   If there are any errors, the error message text appears in the summary control. (Red stars appear and disappear immediately as you work with the fields.)

   > **Note** Required fields are checked the first time only when you submit the form, so that users are not forced to enter data into the form in a specific order. However, after the first time the form is submitted, errors are reported until the field is filled in.

If you are not seeing the errors you expect to see, review the settings for each of the validation controls and try again.

Next Steps

The validation you added in this walkthrough to the Web Forms page illustrates the basic concepts of Web validation controls. There is more you can do with validation controls, both with their basic functionality as well as in code. You could expand on the walkthrough in these ways:

- Check that the user enters a birth date earlier than today by adding a second **CompareValidator** control and setting its operator to **LessThan** and its comparison value to today. You can use a data-binding expression to set the value. Although you are not binding the validator control to data, you are taking advantage of the fact that data binding expressions are evaluated before the control is rendered. The ASP.NET syntax for the control might look like this:

  ```
  <asp:CompareValidator id=c1
  ValueToCompare='<%# System.DateTime.get_Today().ToShortDateString()
   %>'
  ControlToValidate="txtBirthDate" Operator="LessThan" Type="Date" Text="*" runat="server"
  />
  ```

  > **Note** You must call the validator control's **DataBind** method to resolve the expression.

- Check the length of the password to be sure it meets a certain minimum number of characters. You can do this with a **RegularExpressionValidator** control. There is no predefined expression that checks for a minimum number of characters. You can use an expression such as the following, which requires five or more characters:

  ```
  ^[\w]{5,}
  ```

- Test validation in code. Each validation control sets a flag that you can test programmatically. This complements the visual feedback that the controls provide; by testing for valid input in code, you can determine how to proceed (or not proceed) in your application. For details, see Validating Programmatically for ASP.NET Server Controls.

See Also

Introduction to Validating User Input in Web Forms | Web Forms Walkthroughs | Database Security

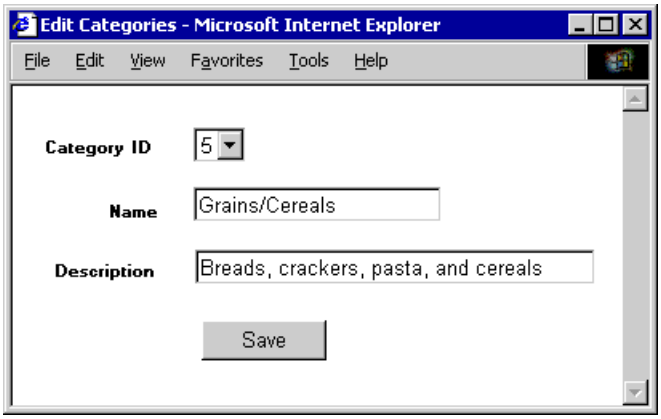
Walkthrough: Updating Data Using a Database Update Query in Web Forms with Visual J#

The Web Forms data-binding architecture allows you to bind control properties to a data source, making it easy to display data on the page. However, the data-binding architecture does not support updates — that is, data binding is one-way in Web Forms pages. Because a majority of Web applications involve reading data and not updating it, one-way data binding keeps the size and complexity of the page architecture to a minimum for the most common scenarios.

However, there are times when you do want to update data. There are several ways to do so. If you are using a dataset, you can write changes to the dataset and then call the data adapter to send changes from the dataset to the database. (For an example of this scenario, see Walkthrough: Using a DataGrid Web Control to Read and Write Data with Visual J#.) Alternatively, you can update the database directly, using SQL statements or stored procedures, which is what this walkthrough illustrates.

> **Note** For details about choosing between these approaches, see Web Data Access Strategy Recommendations.

This walkthrough illustrates how to create a Web Forms page that reads and writes data to a database using SQL statements. For the walkthrough, you will work with the Categories table of the SQL Server Northwind database. In the page, you can select a record, edit it, and then save it to the database. When you are done, your page will look something like the following:



The walkthrough will illustrate a number of concepts:

- Creating command objects that incorporate SQL statements
- Using a SQL statement and a data reader to fetch data from the database
- Creating parameterized queries
- Setting query parameters and executing queries

To complete this walkthrough, you will need:

- Access to a server with the Northwind SQL Server sample database
- Sufficient permissions to be able to create ASP.NET Web application projects on the computer where your Web server is.

The walkthrough is split into a number of smaller pieces:

- Creating the Web Forms page.
- Adding the necessary data components.
- Adding controls to display the data.
- Adding code to read and write data.

Creating the Project and Form

The first step is to create a Web application and a Web Forms page.

To create the project and form

1. On the **File** menu, point to **New**, then click **Project**.
2. In the **New Project** dialog box, do the following:
   a. In the **Project Types** pane, choose **Visual J# Projects**.
   b. In the **Templates** pane, choose **ASP.NET Web Application**.

c. In the **Location** box, enter the complete URL for your application, including http://, the name of the server, and the name of your project. The Web server must have IIS version 5 or later and the .NET Framework installed on it. If you have IIS on your computer, you can specify `http://localhost` for the server.

When you click **OK**, a new Web Forms project is created at the root of the Web server you specified. In addition, a new Web Forms page called WebForm1.aspx is displayed in the Web Forms Designer in Design view.

> **Tip** If you have trouble creating a Web application project, see Web Access Failed Dialog Box.

Adding the Data Components

To access the data, you will need a connection to the database you want to read from. You will also need several command objects — two to read data from the database using SQL Select statements, and a third to update the database with a SQL Update statement. Two of these commands will require parameters, so that you can pass run-time values to the statements before executing them. (The data commands can reference a stored procedure instead of SQL statements, but in this walkthrough you will use statements.)

Creating and Configuring the Connection

To begin, add a connection that points to the database you want to read from.

To create the data connection

1. From the **Data** tab of the Toolbox, drag a **SqlConnection** object onto the Designer surface.

   > **Note** Because you are using SQL Server, you use the **SqlConnection** object, which is optimized for working with SQL Server 7.0 or later. If you are using a different database, you use the **OleDbConnection**, which provides ADO.NET access to any OLE DB-compatible data source.

2. Select the connection and in the Properties window, click the button in the **ConnectionString** property box.

   The list displays any existing connections and the **<New Connection...>** option.

3. Unless you already have a connection to the SQL Server Northwind database, choose **<New Connection...>**.

   The **Data Link Properties** dialog box is displayed.

4. Create a connection pointing to your SQL Server and the Northwind database, and then click **OK**.

   > **Note** You need appropriate read/write permissions on the SQL Server you are using. It is recommended that you specify Windows integrated security when creating the connection. Alternatively, you can specify a user name and password and save that information with the connection, but doing so can compromise security. For more information, see Accessing SQL Server from a Web Application.

   The connection information you specified is saved as the connection string for your connection.

Now you need to add commands. You will need three in all, each with a different query:

- One to get all the category ID values.
- One to get a selected category based on an ID that you provide.
- One to update a category with new values.

To create the data command for getting all category ids

1. From the **Data** tab of the Toolbox, drag an **SqlCommand** object onto the Designer surface.
2. Rename the command **cmdCategoriesAll**.
3. Select the command and in the Properties window, click the button in the **Connection** property, and select the connection you made earlier. (You will need to open the **Existing** node.)
4. In the **CommandText** property, click the ellipsis button to open the **Query Builder** dialog box.
5. Use the **Query Builder** dialog box to construct the following SQL statement:

   ```
   SELECT CategoryID, CategoryName, Description FROM Categories
   ```

6. When you have finished, close the **Query Builder** dialog box.

The SQL statement is filled into the **CommandText** property.

The remaining commands are slightly more complex, in that they require parameters.

To create the data command for getting a single category record

1. Drag a second **SqlCommand** object onto the designer surface and set its **Connection** property to **SqlConnection1**.
2. Name the command **cmdCategoriesById**.
3. In the **CommandText** property, click the ellipsis button to open the **Query Builder** dialog box.
4. Use the **Query Builder** dialog box to construct the following SQL statement:

```
SELECT CategoryID, CategoryName, Description
FROM Categories
WHERE (CategoryID = @categoryid)
```

> **Note** The variable **@categoryid** is a "named variable." The exact name of the variable you use in the Where clause is not important, but you will need to know it later in the walkthrough. The "@" prefix is required for SQL Server named parameters.

The final command includes an **Update** statement for sending changes to the database.

To create the data command for updating a category record

1. Drag a third **SqlCommand** object onto the Designer surface and set its **Connection** property to **SqlConnection1**.
2. Name the command **cmdCategoriesUpdate**.
3. Set the command's **CommandText** property to the following statement:

```
UPDATE Categories
SET CategoryName = @categoryname, Description = @categorydescription WHERE (CategoryID =
@categoryid)
```

You have finished adding data components to the page.

Adding Controls

In the page, you will need a drop-down list to allow users to select a category ID, some text boxes to display and edit data with, and a button to save changes.

To add controls to the form

1. From the **Web Forms** tab of the Toolbox, add the following controls to the form and set the specified properties as indicated:

Control	Purpose	Properties
DropDownList	Allows user to select a category list that you will provide at run time.	ID: **ddlCategoryID** AutoPostBack: **true**
TextBox	Allows users to edit the category name.	ID: **txtCategoryName**
TextBox	Allows users to edit the category descriptio n.	ID: **txtCategoryDescription**
Button	Updates the database with changes.	ID: **btnSave** Text: **Save**

2. Add labels in front of the controls to indicate their function.

> **Tip** If you are working in grid layout mode, you can add literal text using the Flow Layout Panel (HTML Div Controls) from the **HTML** tab of the Toolbox.

Adding Code to Display and Update the Data

To finish the page, you will need to add code to accomplish three tasks:

- When the page is initialized, you need to populate the drop-down list with the category IDs. Since the first category ID will

be automatically selected, you can also display the first category name and description in the text boxes.

- When users select a category ID from the drop-down list, you need to determine which ID they selected. You then use that value to execute the SQL statement that brings back a single record.
- When users click the Save button, you need to get their changes and use them to execute an Update statement with the new values.

Populating the Drop-Down List at Page Initialization

To populate the drop-down list, you will execute a query that returns all the records in the Categories table. You only need to do this the first time the page runs; thereafter, the values in the list are preserved in the control's view state.

After executing the query, you will use a data reader (SqlDataReader object) to fetch the records. The data reader is specifically designed to be a read-only, forward-only reader, so it is optimized for extremely fast data access.

To add code to populate the drop-down list

1. Double-click the Web Forms page to open the Code Editor and find the **Page_Load** event handler.
2. Create code to perform the following functions:
   a. Test the page's **IsPostBack** property to determine if this is the first time the page is running. If so, perform the rest of these steps.
   b. Create an instance variable of type **SqlDataReader**.
   c. Open the connection.
   d. Call the **ExecuteReader** method of the **cmdCategoriesAll** object, which executes the SQL statement in the data command and returns its result set to the data reader object.
   e. Provide a way, such as a Boolean flag, to know when you are fetching the first record, because you want to display the information from the record only.
   f. In a loop, use the data reader to fetch each successive record.
   g. For each record, create a new item in the drop-down list and assign the category ID to it.
   h. For the first record only, get the category name and description and set them to the **Text** property of the text boxes.
   i. Close the data reader.
   j. Close the connection. This is an important step; the connection remains open while the data is being fetched from the database. Therefore, when you no longer need the connection, you want to close it and free the resources it uses.

The following example shows the complete **Page_Load** method with the code required to perform the above steps:

```
// Visual J#
private void Page_Load(Object sender, System.EventArgs e)
{
   // Put user code to initialize the page here
   if (!this.get_IsPostBack())
   {
      // Executes only the first time the page is processed. After
      // that, the list is already in the drop-down list and is
      // preserved in view state.
      System.Data.SqlClient.SqlDataReader dreader;
      sqlConnection1.Open();
      dreader = cmdCategoriesAll.ExecuteReader();
      boolean firstrow = true;
      while (dreader.Read())
      {
         ddlCategoryID.get_Items().Add(new ListItem(dreader.get_Item(0).ToString()));
         if (firstrow)
         {
            txtCategoryName.set_Text(dreader.get_Item(1).ToString());
            txtCategoryDescription.set_Text ( dreader.get_Item(2).ToString());
            firstrow = false;
         }
      }
      dreader.Close();
```

```
        sqlConnection1.Close();
    }
}
```

Displaying a Record When Users Select a Category ID

The drop-down list displays category IDs. When the user selects one — that is, when the control's **SelectedIndexChanged** event is raised — you want to display the corresponding record.

When you created the drop-down list, you set its **AutoPostBack** property to **true**. The result is that the page is posted to the server as soon as the user makes a selection.

To get a category record, you will execute a SQL statement that uses a Where clause to find only the category whose ID the user selected. To pass this value to the statement, you set the value of a parameter (a **SqlParameter** object) that is part of the data command.

To add code to populate the drop-down list

1. Double-click the drop-down list to open the Code Editor and create the **SelectedIndexChanged** event handler.
2. Create code to perform the following functions:
   a. Get the text of the user's selection from the drop-down list.
   b. Set the parameter (@categoryid) of the **cmdCategoriesById** command to the selected ID.
   c. Open the connection.
   d. Create a data reader instance variable and call the **ExecuteReader** method of the **cmdCategoriesById** object to execute the command. In this case, you will pass a flag to the method indicating that you want to return only a single record.
   e. Use the data reader to fetch the returned record and then display its values in the text boxes.
   f. Close the data reader.
   g. Close the connection.

   The following example shows the complete **SelectedIndexChanged** method with the code required to perform the above steps:

```
// Visual J#
private void ddlCategoryID_SelectedIndexChanged (Object sender, System.EventArgs e)
{
   String categoryid;
   categoryid = ddlCategoryID.get_SelectedItem().get_Text();
   ((System.Data.SqlClient.SqlParameterCollection)cmdCategoriesById.get_Parameters()).
      get_Item("@categoryid").set_Value(categoryid);
   sqlConnection1.Open();
   System.Data.SqlClient.SqlDataReader dreader;
   dreader = cmdCategoriesById.ExecuteReader(CommandBehavior.SingleRow);
   if (dreader.Read())
   {
      txtCategoryName.set_Text(dreader.get_Item(1).ToString());
      txtCategoryDescription.set_Text(dreader.get_Item(2).ToString());
   }
   dreader.Close();
   sqlConnection1.Close();
}
```

Updating a Record

Finally, you want to add code that will update the database with changes from the record displayed in the page. As with fetching a single record, the update requires that you set some parameters and then execute a statement (in this case, an Update statement). However, because the Update statement does not return any records, the code for executing the statement is relatively simple.

To update records

1. Double-click the Save button to create a **Click** handler.
2. Create code to perform the following functions:
   a. Set parameters (three in all) to the values of the controls in the page.
   b. Open the connection.
   c. Call the **ExecuteNonQuery** method of the **cmdCategoriesUpdate** command.
   d. Close the connection.

   The following example shows the complete **Click** method with the code required to perform the above steps:

```
// Visual J#
private void btnSave_Click (Object sender, System.EventArgs e)
{
   ((System.Data.SqlClient.SqlParameterCollection) cmdCategoriesUpdate.get_Parameters()).
      get_Item("@categoryid").set_Value(ddlCategoryID.
      get_SelectedItem().get_Text());
   ((System.Data.SqlClient.SqlParameterCollection)cmdCategoriesUpdate.get_Parameters()).
      get_Item("@categoryname").set_Value(txtCategoryName.get_Text());
   ((System.Data.SqlClient.SqlParameterCollection)cmdCategoriesUpdate.get_Parameters()).
      get_Item("@categorydescription").set_Value(txtCategoryDescription.get_Text());
   sqlConnection1.Open();
   cmdCategoriesUpdate.ExecuteNonQuery();
   sqlConnection1.Close();
}
```

Testing

You can now test the form to make sure it displays Categories data and that users can make updates.

To test the page

1. In Solution Explorer, right-click the page and choose **View in Browser**.
2. When the page is displayed, make sure that the drop-down list contains a list of category IDs and that the first category is displayed in the text boxes.
3. Select a different category from the list.

   The new record will be displayed.

   > **Note** If selecting a new category ID in the drop-down list does not change the record, make sure that the control's **AutoPostBack** property is set to **true**.

4. Make a change in one of the text boxes, and then click the **Save** button.
5. Move to a different category record, and then back to the one you changed.

   The changes you made are visible when you redisplay the record.

Next Steps

This walkthrough has illustrated the basic steps involved in using a data reader to fetch records directly from the database and using a data command to make updates. To enhance the Web Forms page you have created, you might want to explore some of the following ideas:

- Provide an alternative way for users to select categories. In this walkthrough, the drop-down list displayed category IDs, which ordinarily are not very useful to users. This was done to allow users to edit the name, if desired, and to maintain clarity for how information from database records was moving into and out of controls.

  A relatively easy change would be to display category names as the text of the drop-down list (the information that is visible in the list) and category IDs as the value of the list. (However, this would not allow users to edit the name.) You could then pass the value as a parameter value instead of passing the text.

- Add error checking. Typically, you want to make your code robust by enclosing all data-access statements in try-catch blocks. Data access is subject to many possible problems (many of them out of the control of your application, such as network problems). For details, see Best Practices for Throwing Exceptions from Components.

- Add concurrency control. In this walkthrough, you simply update a record in the database without determining first whether another user might have also made changes. For details, see Concurrency Control in ADO.NET.

See Also

Walkthrough: Creating Read-Only Data Access in a Web Forms Page with Visual J# |
Introduction to ADO.NET Connection Design Tools | Introduction to DataCommand Objects in Visual Studio |
Executing a Data Command That Returns a Result Set | Executing Updates or Database Commands using a Data Command |
Setting and Getting Data Command Parameters | Concurrency Control in ADO.NET | Web Forms Walkthroughs

Walkthrough: Using a DataGrid Web Control to Read and Write Data with Visual J#

The architecture of Web Forms data binding makes it easy to display data in page controls. However, data binding is not two-way — that is, data binding reads data from a data source but does not update it. Updating is more complex than displaying data, and because the majority of Web Forms pages do not require that data be written back to the source, Web Forms data binding keeps page sizes and page processing to a minimum by not including update code.

Of course, there are times when you do want to create a Web Forms page that updates data. This walkthrough illustrates one way to do that. It shows how you can use a **DataGrid** control to display data, allow users to edit it, and then send the changed data back to the source.

When the page is running, it will look something like the following:

	CategoryID	CategoryName	Description
Edit	1	Beverages	Soft drinks, coffees, teas, beers, and ales
Edit	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
Edit	3	Confections	Desserts, candies, and sweet breads
Edit	4	Dairy Products	Cheeses
Edit	5	Grains/Cereals	Breads, crackers, pasta, and cereal
Edit	6	Meat/Poultry	Prepared meats
Edit	7	Produce	Dried fruit and bean curd
Edit	8	Seafood	Seaweed and fish

To complete this walkthrough, you will need:

- Access to a server with the Northwind SQL Server sample database.
- Sufficient permissions to be able to create ASP.NET Web application projects on the computer where your Web server is.

The walkthrough is split into a number of smaller pieces:

- Creating the Web Forms page.
- Adding the necessary data components.
- Adding the **DataGrid** control to display the data.
- Adding code to read data from the database and bind the grid to the data.
- Configuring the **DataGrid** control to allow users to edit data.
- Adding the code to update the data.

Creating the Project and Form

The first step is to create a Web application and a Web Forms page.

To create the project and form

1. On the **File** menu, point to **New**, then click **Project**.
2. In the **New Project** dialog box, do the following:
   a. In the **Project Types** pane, choose **Visual J# Projects**.
   b. In the **Templates** pane, choose **ASP.NET Web Application**.
   c. In the **Location** box, enter the complete URL for your application, including http://, the name of the server, and the name of your project. The Web server must have IIS version 5 or later and the .NET Framework installed on it. If you have IIS on your computer, you can specify `http://localhost` for the server.

      When you click **OK**, a new Web Forms project is created at the root of the Web server you specified. In addition, a new Web Forms page called WebForm1.aspx is displayed in the Web Forms Designer in Design view.

      Tip If you have trouble creating a Web application project, see Web Access Failed Dialog Box.

Creating and Configuring a Dataset

In Web Forms pages, you have several choices for how you can access data. One is to use a dataset, which is an in-memory data cache. Alternatively, you can access a database directly, using data commands to execute SQL statements or stored procedures. As a general rule, it is easier to update data if you use a dataset, and that is what you will use in this walkthrough. For more

information, see Web Data Access Strategy Recommendations.

You do not directly add a dataset to a page. Instead, you will perform the following set of steps:

1. Create a data adapter using a wizard. The adapter contains SQL statements used to read and write database information. The wizard helps you define the SQL statements you need. If necessary, the wizard also creates a connection to the database.
2. Generate the dataset schema. In this process, you will have Visual Studio create a new dataset class based on the tables and columns you are accessing. When you generate the dataset class, you will also add an instance of it to the form.

It is important that you follow all the procedures in this section. Otherwise your page will not have the dataset that you will be using in subsequent parts of the walkthrough.

For an overview of data adapters, see Introduction to Data Adapters. For an overview of datasets, see Introduction to Datasets.

Configuring a Data Connection and Data Adapter

To begin, you create a data adapter that contains the SQL statement used to populate the dataset later. As part of this process, you define a connection to access a database. You configure the data adapter using a wizard, which makes it easy to create the SQL statements you need for data access.

> **Note** When the wizard is done, you must continue to the next section in order to generate a dataset and complete the data access portion of your page.

To create the data connection and data adapter

1. From the **Data** tab of the Toolbox, drag a **SqlDataAdapter** object onto the page.

   > **Note** If you are not working with SQL Server, you use an adapter of type **OleDbDataAdapter**, which provides access to any OLE DB-compatible data source.

   The **Data Adapter Configuration Wizard** starts, which will help you create both the connection and the adapter.

2. In the wizard, do the following:
   a. In the second page, create or choose a connection pointing to the SQL Server Northwind database. For information about getting access to the database, see your database administrator.

      > **Note** You need appropriate read/write permissions on the SQL Server you are using. It is recommended that you specify Windows integrated security when creating the connection. Alternatively, you can specify a user name and password and save that information with the connection, but doing so can compromise security. For more information, see Accessing SQL Server from a Web Application.

   b. In the third page, specify that you want to use a SQL statement to access the database.
   c. In the fourth page, create the following SQL statement:

      ```
      SELECT CategoryID, CategoryName, Description
      FROM Categories
      ```

      For assistance building the SQL statement, click **Query Builder** to launch the **Query Builder** dialog box.

      > **Note** In this walkthrough, you will populate the dataset with all the rows from the categories table. In production applications, you typically optimize data access by creating a query that returns only the columns and rows you need. For an example, see Walkthrough: Displaying Data in a Windows Form Using a Parameterized Query with Visual J#.

   d. Click **Finish**.

      The wizard creates a connection (**sqlConnection1**) containing information about how to access your database. You will also have a data adapter (**sqlDataAdapter1**) that contains a query defining what table and columns in the database you want to access.

After the wizard is complete, you need to generate the dataset based on the SQL query that you created during this procedure. For details, see the next section.

Creating the Dataset

After you have established the means to connect to the database and specified the information you want (via the SQL command in the data adapter), you can have Visual Studio create a dataset. Visual Studio can generate the dataset automatically based on the query you specified for the data adapter. The dataset is an instance of the **DataSet** class based on a corresponding schema (.xsd file) that describes the class's elements (table, columns, and constraints). For details about the relationship between datasets and schemas, see Introduction to Data Access with ADO.NET.

To generate a dataset

1. From the **Data** menu, choose **Generate DataSet**.

   > **Tip** If the **Generate DataSet** command is not enabled, click the page; the page must have focus for the command to appear.

   The **Generate Dataset** dialog box appears.

2. Select the **New** option and name the dataset **dsCategories**.

   In the list under **Choose which table(s) to add to the dataset**, make sure the Categories table is selected.

3. Make sure **Add this dataset to the designer** is checked, and then click **OK**.

   Visual Studio generates a typed dataset class (**dsCategories**) and a schema that defines the dataset. You will see the new schema (**dsCategories.xsd**) in Solution Explorer.

   > **Tip** In Solution Explorer, click the **Show All Files** toolbar button to see the schema file's dependent .jsl file, which contains the code that defines your new dataset class.

   Finally, Visual Studio adds an instance of the new dataset class (**dsCategories1**) to the page.

At this point you have set up everything you need in order to get information out of database and into a dataset.

Adding a DataGrid Control to Display the Data

In this walkthrough, you will add a single control — a **DataGrid** control — that can display all the records from the dataset at the same time and allows you to add the ability to edit records.

The data grid must be bound to the dataset in order to display the data.

To add a bound DataGrid control to the form

1. If you have not already done so, switch to the Web Forms Designer by clicking the tab at the top of the current window.
2. From the **Web Forms** tab of the Toolbox, drag a **DataGrid** control onto the form.
3. Select the control, press F4 to display the Properties window, and at the bottom of the window, click **Property Builder**.

   The **DataGrid Properties** dialog box appears.

4. In the General tab, make the following settings:

Property	Setting	Description
DataSource	dsCategories1	Binds the grid to the dataset.
DataMember	Categories	Specifies that the grid should display data from the categories table in the dataset
Data key field	CategoryID	Specifies that the primary key for categories record is the CategoryID column. This will allow you later to identify which record to update in the dataset.

5. Click **OK** to close the **DataGrid Properties** dialog box.
6. If you want to change the appearance of the grid, set the **Font**, **BackColor**, and other properties.

   > **Tip** An easy way to set the grid's appearance is to click **Auto Format** at the bottom of the Properties window and select a predefined look.

Filling the Dataset and Displaying Data in the DataGrid Control

Although the grid is bound to the dataset you created, the dataset itself is not automatically filled in. Instead, you must fill the dataset yourself by calling a data-adapter method. For details about filling datasets, see Introduction to Datasets.

Even after the dataset is filled in, the **DataGrid** control does not automatically display the data. You must explicitly bind the grid to its data source. For more information, see Introduction to Data Binding in Web Forms Pages.

To fill the dataset and display data in the DataGrid control

1. Double-click the page to display the page's class file in the Code Editor.
2. In the **Page_Load** event handler, call the data adapter's **Fill** method, passing it the dataset you want to populate:

```
// Visual J#
sqlDataAdapter1.Fill(dsCategories1);
```

3. Call the **DataGrid** control's **DataBind** method to bind the control to the dataset. You do not want to rebind the control each time the page makes a round trip, however, because if you do, you will lose changes that the user has made in the grid. You should therefore bind the grid only at these times:

   - The first time the page is called.
   - When the dataset has changed.

   For now, you want to bind the grid the first time the page is called, which you can do by testing the page's **IsPostBack** property. Add code to the **Page_Load** event handler after the call to the adapter's **Fill** method. The complete handler will look like the following:

```
// Visual J#
private void Page_Load(Object sender, System.EventArgs e)
{
   // Put user code to initialize the page here
   sqlDataAdapter1.Fill(dsCategories1);
   if (! get_IsPostBack())
   {
      DataGrid1.DataBind();
   }
}
```

Adding the Ability to Edit Data

As configured, the **DataGrid** control will display information from the Categories table. But you also want users to be able to edit individual rows in the grid. To do this, you add an Edit button to each row of the grid. The user clicks the button and the grid is redisplayed in edit mode, with text boxes where users can edit the individual columns. When the row is in edit mode, the Edit button is replaced with two other buttons, an Update button and a Cancel button, as shown in the following figure:

	CategoryID	CategoryName	Description
Update Cancel	1	Beverages	Soft drinks, coffees, teas,
Edit	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
Edit	3	Confections	Desserts, candies, and sweet breads
Edit	4	Dairy Products	Cheeses
Edit	5	Grains/Cereals	Breads, crackers, pasta, and cereal
Edit	6	Meat/Poultry	Prepared meats
Edit	7	Produce	Dried fruit and bean curd
Edit	8	Seafood	Seaweed and fish

You can set properties of the **DataGrid** control to display these buttons. However, the buttons are not hooked up automatically; instead, when an Edit or Cancel or Update button is clicked, it raises an event that you respond to. You will need to add short event handlers that do the following:

- Edit button – set the current row into edit mode.
- Cancel button – return the row to display mode.

The event handler for the Update button is more involved, since it performs the update that is the heart of this walkthrough. You will read about this in the next section.

To add editing capability to the DataGrid control

1. In Design view of the Web Forms Designer, select the **DataGrid** control, press F4 to display the Properties window, and then

click **Properties Builder** at the bottom of the window.

The **DataGrid Properties** dialog box is displayed.

2. Click the **Columns** tab.
3. Under **Column list**, scroll down in the **Available columns** list and open the **Button Column** node.
4. Select **Edit, Update, Cancel** and click the add ( )button to add the buttons to the **Selected columns** box.
5. Click **OK**.

   The **DataGrid** control is redisplayed with an Edit link button in the left-hand column. (You do not initially see the Update and Cancel buttons.)

Now that you have the Edit button, you need to create the event handlers to set edit mode for the rows. To control edit mode, you set the **DataGrid** control's **EditItemIndex** property to the index (zero-based) of the row to edit. For example, to put the third row into edit mode, you set the property to 2. To return a row to display mode, you set the property to –1. After changing edit mode, you must rebind the grid so it displays the data in the row.

You can determine what row the user is in from the event-object passed to the handler. The event object for these events contains an **Item** property that represents the entire **DataGrid** row being updated. The **Item** object in turn supports various properties, including the **Item.ItemIndex** property, which returns the index value of the row you are working with.

To set edit modes

1. In Design view, select the grid and then press F4 to open the Properties window.
2. Click the Events button ( ) at the top of the Properties window.
3. Double-click **EditCommand** in the grid.

   A **DataGrid1_EditCommand** handler is created.

4. Repeat steps 2 and 3 for the **CancelCommand** event.

   A **DataGrid1_CancelCommand** handler is created.

5. In the **EditCommand** event handler, add the following code:

```
// Visual J#
DataGrid1.set_EditItemIndex(e.get_Item().get_ItemIndex());
DataGrid1.DataBind();
sqlConnection1.Close();
```

6. In the **CancelCommand** event handler, add the following code:

```
// Visual J#
DataGrid1.set_EditItemIndex(-1);
DataGrid1.DataBind();
sqlConnection1.Close();
```

Testing What You Have So Far

You now have a **DataGrid** control that displays data from a dataset (after the dataset is filled from the database) and you have configured the grid with an Edit button for each row. Even though the page is not yet complete, it is useful to test it now to make sure that the edit capability is working correctly.

To test what you have so far

1. In Solution Explorer, right-click your Web Forms page and choose **View in Browser**.

   The project is compiled and the Web Forms page appears in a browser pane in the designer. If everything is working correctly, the grid will be filled with data and the first column in the grid will contain links that read "Edit."

2. Click the **Edit** link for any row in the grid.

   The row is redisplayed with these changes:

   - The Edit link is replaced with Update and Cancel links.

- The data is redisplayed in **TextBox** controls.

3. Click **Cancel**.

   The row is redisplayed in its original form.

The next section details how you can have the Update link write changes from the grid to the dataset and database.

Updating the Dataset and Database

Up to this point, the **DataGrid** control has done most of the work involved in simply displaying data. However, as with other ASP.NET server controls, the grid does not include automatic provisions for updates — that is, for taking changes made by a user in the grid and sending them back to the data source. To perform updates, you will need to write some code.

The update will actually occur in two stages. First, you must update the dataset with changes made in the grid. Then you must write changes from the dataset back to the database. For more information, see Introduction to Dataset Updates.

When a grid row is in edit mode, the leftmost column contains an Update link. When users click the link, it raises an **UpdateCommand** event. You will write all your update code in the handler for this event.

To create the UpdateCommand handler

1. In Design view, select the grid and then press F4 to open the Properties window.
2. Click the Events button ( ) at the top of the Properties window.
3. Double-click **UpdateCommand** in the grid.

   A **DataGrid1_UpdateCommand** event handler is created.

Steps for Updating From the DataGrid Control

The outline for what you will be doing is this:

1. Determine what row (by index) in the DataGrid control was updated. Then get the datakey out of that grid row so that you identify what row (by ID) is being updated.
2. Get the changed values out of the grid row that the user updated.
3. Use the datakey value to find the corresponding row in the dataset table and then write the changes to that row. At this point, you have updated the dataset, but not yet the database itself.
4. Send changes from the dataset to the database. This executes a SQL command or stored procedure that copies changes from the dataset to the database.
5. Refresh the contents of the **DataGrid** control.
6. The following sections explain the details of each of these steps. If you prefer, you can skip these explanations and go directly to the code that follows.

To update from the DataGrid control

1. Identify what **DataGrid** row was updated by getting the **ItemIndex** property of the row (**Item** object) passed in the event object. Then use the index value to get the corresponding value out of the grid's **DataKeys** collection:

```
// Visual J#
String key = DataGrid1.get_DataKeys().get_Item(e.get_Item().get_ItemIndex()).ToString();
```

2. Get the changed values out of the **DataGrid** row. To do so:
   a. From the item passed in the event object, get the appropriate cell (zero-based) out of the **Cells** collection from the item passed in the event object. For example, the left-most column in the grid is **Cells(0)**.
   b. For each cell, get its **Controls** collection, which contains all the elements that appear in the cell.
   c. Get the first (and only) control out of the collection — in this case, a **TextBox** control. To get the **TextBox**, declare a local variable of type **TextBox** and cast the object in the Controls collection to it.
   d. Get the **TextBox** control's value (its **Text** property).

   The following example shows how to perform these steps.

```
// Visual J#
String categoryname;
```

```
String categoryDescription;
TextBox tb;
tb = (TextBox)(((System.Web.UI.WebControls.TableCell)(e.get_Item().get_Cells().get_Item(2
))).get_Controls().get_Item(0));
categoryName = tb.get_Text();
tb = (TextBox)(((System.Web.UI.WebControls.TableCell)(e.get_Item().get_Cells().get_Item(3
))).get_Controls().get_Item(0));
categoryDescription = tb.get_Text();
```

3. Find the corresponding row in the data table. The typed **dsCategories** dataset contains a special **FindBy** method — in this case, the **FindByCategoryID** method — that locates a row by its primary key and returns a reference to it. Create a variable for the typed data row and call the method:

```
// Visual J#
dsCategories.CategoriesRow r;
r = dsCategories1.get_Categories().FindByCategoryID(Integer.parseInt(key));
```

4. Update the row by changing values in the row you located in Step 3, as in the following example:

```
// Visual J#
r.set_CategoryName(categoryName);
r.set_Description(categoryDescription);
```

5. Send changes from the dataset to the database by calling the data adapter's **Update** method:

```
// Visual J#
sqlDataAdapter1.Update(dsCategories1);
DataGrid1.DataBind();
```

6. Switch the current row in the grid out of editing mode.

```
// Visual J#
DataGrid1.set_EditItemIndex(-1);
```

7. Data-bind the **DataGrid** control:

```
// Visual J#
DataGrid1.DataBind();
```

The following code shows what the completed **UpdateCommand** event handler looks like. Copy this code and paste it into the class file for the Web Forms page.

> **Tip** Be sure to overwrite the skeleton event handler you created earlier; otherwise you will have two methods with identical names and signatures.

```
// Visual J#
private void DataGrid1_UpdateCommand (Object source, System.Web.UI.WebControls.DataGridCo
mmandEventArgs e)
{
   String categoryName, categoryDescription;

   // Gets the value of the key field of the row being updated
   String key = DataGrid1.get_DataKeys().get_Item(e.get_Item().get_ItemIndex()).ToString(
);

   // Gets get the value of the controls (textboxes) that the user
   // updated. The DataGrid columns are exposed as the Cells collection.
   // Each cell has a collection of controls. In this case, there is only one
   // control in each cell -- a TextBox control. To get its value,
```

```
    // you copy the TextBox to a local instance (which requires casting)
    // and extract its Text property.
    //
    // The first column -- Cells(0) -- contains the Update and Cancel buttons.
    TextBox tb;

    // Gets the value the TextBox control in the third column
    tb = (TextBox)(((System.Web.UI.WebControls.TableCell)(e.get_Item().get_Cells().get_Ite
m(2))).get_Controls().get_Item(0));
    categoryName = tb.get_Text();

    // Gets the value the TextBox control in the fourth column
    tb = (TextBox)(((System.Web.UI.WebControls.TableCell)(e.get_Item().get_Cells().get_Ite
m(3))).get_Controls().get_Item(0));
    categoryDescription = tb.get_Text();

    // Finds the row in the dataset table that matches the
    // one the user updated in the grid. This example uses a
    // special Find method defined for the typed dataset, which
    // returns a reference to the row.
    dsCategories.CategoriesRow r;
    r = dsCategories1.get_Categories().FindByCategoryID(Integer.parseInt(key));

    // Updates the dataset table.
    r.set_CategoryName(categoryName);
    r.set_Description(categoryDescription);

    // Calls a SQL statement to update the database from the dataset
    sqlDataAdapter1.Update(dsCategories1);

    // Takes the DataGrid row out of editing mode
    DataGrid1.set_EditItemIndex(-1);

    // Refreshes the grid
    DataGrid1.DataBind();
    // Close the connection
sqlConnection1.Close();
}
```

Testing

You are now finished. To illustrate how the grid works and make sure it is updating data properly, you should test the page.

To test the page

1. In Solution Explorer, right-click your Web Forms page and choose **View in Browser**.

   The project is compiled and the Web Forms page appears in a browser pane in the designer.

2. Click the **Edit** link for any row in the grid, and edit the row using the text boxes.
3. Click **Update**.

   The grid is redisplayed with your changes. If you examine the database, you will see that the change has been written there are well.

Next Steps

In this walkthrough, you used a **DataGrid** control on a Web Forms page not just to view data, but also to edit it. For the sake of clarity, the walkthrough skipped some details that you would likely want to think about in a production application. Some of the areas in which you might want to enhance what you have learned in this walkthrough are:

Enhancing the DataGrid Control

This walkthrough did not take advantage of some of the DataGrid control's more advanced features. Some of the enhancements you could make are:

- Configure individual columns in the grid by choosing specific columns to display, changing column headers and so on. For details, see Adding Bound Columns to a DataGrid Web Server Control and Specifying Grid Item Format in a DataGrid Web Server Control
- Allow users to delete rows. For details, see Allowing Users to Delete Items in a DataGrid Web Server Control. In addition to specifying a Delete button, you would need to add code to remove records from the dataset. For details, see Deleting Records in a Dataset.
- Use paging. In this instance, you only displayed a few records. However, it is common to display many more, and if so, it is desirable to allow the grid to display information a page at a time. For details, see Paging Behavior in DataGrid Web Server Controls.

Caching the Dataset

In the walkthrough, the dataset is reinstantiated and refilled from the database each time the page makes a round trip to the server. This strategy, while straightforward to implement, can result in unnecessary traffic to your database server.

An alternative is to store the dataset after it is filled and each time it is changed. Then, instead of rereading the database information, you can restore the dataset with each round trip. There are a variety of options for storing the dataset, including in session state (on the server) or in view state (on the client, in the page). For details about what strategy to use, see State Management Recommendations.

For example, if you want to store the dataset in session state, you might include code such as the following in the **Page_Load** event handler:

```
// Visual J#
   if (this.get_Session().get_Item("mydatset")== null)
   {
      sqlDataAdapter1.Fill(dsCategories1);
      this.get_Session().set_Item("mydataset",dsCategories1);
   }
   else
   {
      dsCategories1 = (dsCategories) this.get_Session().get_Item("mydatset");
   }
   if (! get_IsPostBack())
   {
         DataGrid1.DataBind();
   }
```

Note that if you store a dataset in Session or Application state, you need to cast it to the appropriate dataset type when you get it back out.

Be sure to refresh the copy of the dataset in session state each time it changes. For example, you might add the following line in the **UpdateCommand** event handler immediately after you update the database:

```
// Visual J#
sqlDataAdapter1.Update(dsCategories1);
this.get_Session().set_Item("mydataset", dsCategories1);
```

Using Direct Access to the Database instead of a Dataset

In the walkthrough, you created a dataset and bound the grid to it. To update the database, you first updated the dataset, then used the data adapter's **Update** method to send changes to the database.

Using a dataset makes it easy to perform updates, but at the cost of some memory and a very slight performance overhead. As an alternative, you can use data commands (OleDbCommand Class or SqlCommand Class objects) to execute SQL statements or stored procedures directly. For example, you can configure a data command with a parameterized SQL Update statement to update the Categories table in the database. After getting the changes from the grid, you set parameters for the Update statement and then execute it.

For more information, see Introduction to DataCommand Objects in Visual Studio.

Validating Data

In a production application, you would probably include validation checks to be sure that the changes entered by a user were valid. For details, see Data Validation in Datasets.

Concurrency Control

In the walkthrough, you sent updates to the database without worrying about whether the record you were updating had been changed by another user. However, in production applications, you will usually want to include error checking to be sure that you maintain the integrity of your database. For details, see Concurrency Control in ADO.NET.

See Also

Walkthrough: Displaying Data in a Web Forms Page with Visual J# |
Walkthrough: Creating Read-Only Data Access in a Web Forms Page with Visual J# |
Walkthrough: Creating a Distributed Application with Visual J# |
Walkthrough: Updating Data Using a Database Update Query in Web Forms with Visual J# |
Walkthrough: Validating User Input in a Web Forms Page with Visual J# | Web Forms Walkthroughs

Walkthrough: Exploring Event Logs, Event Sources, and Entries with Visual J#

This walkthrough will take you through the major feature areas of event logging in your Visual Studio application. During this walkthrough, you will learn how to:

- Create an **EventLog** component.
- Write code to create and delete custom event logs.
- Write entries of various types to the custom log.
- Read entries from the custom log.
- Verify the existence of logs and event sources.
- Clear log entries.
- Use Server Explorer to verify the results of your event log actions.

Creating the User Interface

In this walkthrough, you will create a Windows Form and use a series of controls on it to launch a series of event-logging actions.

To create the form and controls for your application

1. From the **New Project** dialog box, create a **Visual J# Windows Application**, and name it **EventLogApp1**.
2. Add eight buttons to the Windows Form, and set the following properties for them:

Control	Text Property	Name Property
Button1	**Create Custom Log**	**CreateLog**
Button2	**Delete Log**	**DeleteLog**
Button3	**Write Entry**	**WriteEntry**
Button4	**Clear Log**	**ClearLog**
Button5	**Verify Log Exists**	**VerifyLog**
Button6	**Verify Source Exists**	**VerifySource**
Button7	**Remove Event Source**	**RemoveSource**
Button8	**Read Entry**	**ReadEntry**

3. In turn, do the following for each button:
   a. In the designer, double-click the button to create a default event handler for that button. The Code Editor appears and a stub for the button's **Click** event appears.
   b. Return to Design view, and double-click the next button.
   c. Continue until you have created a default event handler stub for each button.
4. From the **Components** tab of the Toolbox, drag an **EventLog** component to the form.

   An **EventLog** component instance appears in the component tray area at the bottom of the form.

Creating and Deleting a Custom Log

In this procedure, you will use the **SourceExists** method to verify that the source you are using does not already exist. If it does exist, you will delete it, and then you will call the **CreateEventSource** method. The system will create a custom log for you when this code is run.

To create the custom log

1. In the Code Editor, add an **import** statement to the beginning of the file to reference the **System.Diagnostics** namespace.

```
// Visual J#
import System.Diagnostics.*;
```

2. Locate the **CreateLog_Click** procedure.
3. Type in the following code:

```
// Visual J#
if (!(EventLog.SourceExists("Source1")))
```

```
{
   EventLog.CreateEventSource("Source1", "NewLog1");
}
else
{
   EventLog.DeleteEventSource("Source1");
   EventLog.CreateEventSource("Source1", "NewLog1");
}
```

To delete a custom log

1. In the Code Editor, locate the **DeleteLog_Click** procedure.
2. Type in the following code:

```
// Visual J#
if (EventLog.SourceExists("Source1"))
{
   EventLog.Delete("NewLog1");
}
```

Writing Entries to the Log

In this procedure, you will use the **EventLog** component instance you created to write entries to the log. To do so, you will first configure the component to use the source string you just created, then you will specify two entries to write: an informational event and an error event.

To write entries to the log

1. In the Code Editor, locate the **WriteEntry_Click** procedure.
2. Set the **Source** property for the **EventLog** component instance to Source1, which is the source string you used when you created your custom log. Use this code:

```
// Visual J#
eventLog1.set_Source("Source1");
```

3. Call the **WriteEntry** method and specify a text message. Use this code:

```
// Visual J#
eventLog1.WriteEntry("This is an informational message");
```

4. Call the **WriteEntry** method again and use the overloaded form to specify both a message and a type of error. Use this code:

```
// Visual J#
eventLog1.WriteEntry("This is an error message", EventLogEntryType.Error);
```

Clearing Log Entries

In this procedure, you will use the **Clear** method to remove existing entries from the custom log.

To clear log entries

1. In the Code Editor, locate the **ClearLog_Click** procedure.
2. Call the **Clear** method on the **EventLog** component instance:

```
// Visual J#
eventLog1.Clear();
```

Verifying Logs and Sources

In this procedure, you will create two procedures: one that verifies the existence of the custom log, and one that verifies the existence of your source string. These procedures will be used to test the results of various actions you perform when you run the project.

To verify that the custom log exists

1. In the Code Editor, locate the **VerifyLog_Click** procedure.
2. Create a message box that evaluates whether the specified event log exists and displays true or false accordingly. Use this code:

```
// Visual J#
MessageBox.Show("" + EventLog.Exists("NewLog1"));
```

To verify that the source exists

1. In the Code Editor, locate the **VerifySource_Click** procedure.
2. Create a message box that evaluates whether the specified source exists and displays true or false accordingly. Use this code:

```
// Visual J#
MessageBox.Show("" + EventLog.SourceExists("Source1"));
```

Removing Sources

In this procedure, you will write code to delete a source string. To do so, you will first verify that the source in question (Source1) exists, then call the **DeleteEventSource** method to remove it.

To remove the event source you created

1. In the Code Editor, locate the **RemoveSource_Click** procedure.
2. Add the following code:

```
// Visual J#
if (EventLog.SourceExists("Source1"))
{
   EventLog.DeleteEventSource("Source1");
}
```

Reading Entries

In this procedure, you will write code to iterate through the event log's entries collection and display the existing messages in the log.

To read entries from the custom log you created

1. In the Code Editor, locate the **ReadEntry_Click** procedure.
2. Add the following code:

```
// Visual J#
int count = eventLog1.get_Entries().get_Count();
for (int i = 0; i < count; i++)
{
   MessageBox.Show(eventLog1.get_Entries().get_Item(i).get_Message());
}
```

Testing your Code

In this section, you will use Server Explorer to verify the results of your code.

Launching Server Explorer

1. From the **View** menu, access Server Explorer.

2. Expand the node for your current server, and then expand the **Event Logs** node beneath it.

Build and run your application

1. Save your files and press F5 to build and launch your project. The form appears with the eight buttons you created.
2. Click the **Create Custom Log** button.

   > **Note** You must have appropriate privileges to the server on which your application runs in order to create Windows event logs. If you receive a security error at this point, see your system administrator.

3. Return to the product, while still in run mode, and right-click the **Event Logs** node in Server Explorer.
4. Click **Refresh**.
5. Verify that the **NewLog1** log now appears in the **Event Logs** node.

To test creating, deleting, and verifying custom logs

1. Return to your running form, and click the **Verify Log Exists** button.

   A prompt should appear, displaying "true".

2. Click the **Delete Log** button, and then click the **Verify Log Exists** button again.

   This time, the prompt should say "false".

3. Click the **Create Custom Log** button again to recreate the log.

To test writing entries to and reading entries from the custom log

1. In the form, click the **Write Entry** button.
2. Access Server Explorer, and expand the **NewLog1** log.
3. Expand the **Source1** node beneath it.

   You should now see that two entries have been added to the log. One will have an icon indicating that it is an informational entry, and one will have an icon indicating that it is an error.

4. Return to the form, and click the **Read Entry** button.

   You will receive two prompts: one containing the informational entry and one containing the error entry.

   > **Note** If you clicked the **Write Entry** button multiple times, you will have received more prompts.

To test clearing the log

1. In the form, click the **Clear Log** button.
2. In Server Explorer, right-click the **NewLog1** node, and then click **Refresh**.

   You should now see that the log no longer contains entries.

To test removing the source string

1. In the form, click the **Remove Event Source** button.
2. Click the **Verify Source Exists** button. You should receive a prompt that says "False" to indicate that the source **Source1** no longer exists.
3. Click the **Write Entry** button.

   > **Note** This action will allow you to write entries to the log, because the **WriteEntry** method will set the source if it does not currently exist.

4. In Server Explorer, right-click the **NewLog1** node, and then click **Refresh**. You should see two entries in the log.
5. Click the **Verify Source Exists** button again. You should receive a prompt that says "true" to indicate that the source **Source1** exists.
6. Optionally, you may want to click the **Remove Event Source** button when you are through testing. This will remove the source **Source1** from the registry so that there are no permanent changes to your system's configuration.

See Also

Walkthrough: Installing an Event Log Component with Visual J#

The procedures in this topic walk you through the process of setting up an installation component for an instance of the **EventLog** component, including creating the component instance, adding an installer for it, and compiling and running the installer.

To create your EventLog component

1. On the **File** menu, point to **New**, and then click **Project**.
2. In the **New Project** dialog box, choose **Visual J#** in the left pane, then choose the **Windows Application** template. Name the project **MyEventLog**.
3. From the **Components** tab of the Toolbox, drag an EventLog component to the form.
4. In the Properties window, set the following values:
   - Set the **Log** property to **Application**.
   - Set the **MachineName** property to the name of the server on which the event log resides. Use a period (**.**) for the local computer.
   - Set the **Source** property to the string of your choice. In this case, you can use the name of your project.
5. Save your files.

   Note For more information on creating **EventLog** component instances and setting their properties, see Logging Application, Server, and Security Events.

To create an installation component for your EventLog component

1. In the designer, click the **eventLog1** component.
2. In the **Description** area at the bottom of the Properties window, click the **Add Installer** link.

   A **ProjectInstaller** class appears in your project, and an installation component is added to it. You can now install and run your application.

To build your installation

- Save your project and build it.

  Note Because this installer will function correctly without changing the existing methods, you do not need to override the **Install**, **Commit**, **Rollback**, and **Uninstall** methods.

A setup project will install the compiled project files and run the installers needed to run a Windows application.

To create a setup project for your service

1. On the **File** menu, point to **Add Project**, then click **New Project**.
2. In the Project Types pane, select the **Setup and Deployment Projects** folder.
3. In the Templates pane, select **Setup Project**. Name the project **MyEventLogSetup**.

   A setup project is added to the solution. Next you will add the output from the Windows application project to the setup.

To add the project output to the setup

1. In Solution Explorer, right-click **MyEventLogSetup**, point to **Add**, then choose **Project Output**.

   The **Add Project Output Group** dialog box appears.

2. Verify that **MyEventLog** is selected in the **Project** box.
3. In the list box, select **Primary output**, then click **OK**.

   A project item for the primary output of **MyEventLog** is added to the setup project. Now add a custom action to install the MyNewService.exe file.

To add a custom action to the setup

1. In Solution Explorer, right-click the setup project, point to **View**, then choose **Custom Actions**.

   The **Custom Actions** editor appears.

2. In the **Custom Actions** editor, right-click the **Custom Actions** node and choose **Add Custom Action**.

   The **Select Item in Project** dialog box appears.

3. Double-click the **Application Folder** in the list box to open it, select **Primary output from MyEventLog (Active)**, and click OK.

   The primary output is added to all four nodes of the custom actions: **Install**, **Commit**, **Rollback**, and **Uninstall**.

4. Build the setup project.
5. Browse to the directory where the setup project was saved, and run the .msi file to install **MyEventLog.exe**.

See Also

Introduction to Installation Components | Launching Event Viewer | Logging Application, Server, and Security Events | System Monitoring Walkthroughs | Framework Services Component Walkthroughs

Walkthrough: Retrieving Categories and Counters with Visual J#

The procedures in this walkthrough take you through the process of creating and configuring **PerformanceCounterCategory** component instances and using them to retrieve lists of performance counter categories and counters on your system. A performance counter is the means by which Windows collects performance data on various system resources. Windows contains a set of predefined counters, organized into categories, with which you can interact. Each category and counter is related to a specific area of system functionality.

In this walkthrough you will:

- Instantiate a **PerformanceCounterCategory** component and configure it to interact with a specific category of system-generated counters.
- Create a Windows application that displays information about categories and counters in a list box.
- Use the **GetCategories** method to return a list of categories on the local machine.
- Use the **GetCounters** method to return a list of counters from the specified category.

To create your Windows application

1. From the **New Project** dialog box, create a **Visual J# Windows Application**. Name the project **PerformanceCounterCategoryExample**.
2. From the **Windows Forms** tab of the Toolbox, add two buttons and two list boxes to your form. Arrange them in any order you like, and then set the following properties for them:

Control	Property	Value
Button1	Name	**btnGetCounters**
	Text	**Get Counters**
Button2	Name	**btnGetCategories**
	Text	**Get Categories**
ListBox1	Name	**lstCounters**
	ScrollAlwaysVisible	**True**
ListBox2	Name	**lstCategories**
	ScrollAlwaysVisible	**True**

3. Save your work.

To retrieve the category list

1. In Design view, double-click the **Get Categories** button to access the Code Editor. Your cursor will be positioned in the **btnGetCategories_Click** event.
2. At the top of the screen, add an **import** statement referencing the **System.Diagnostics** namespace.

```
// Visual J#
import System.Diagnostics.*;
```

3. In the **btnGetCategories_Click** procedure, add the following code to retrieve the list of categories from the local machine.

```
// Visual J#
private void btnGetCategories_Click (Object sender, System.EventArgs e)
{
   PerformanceCounterCategory[] myCat2;

   // Remove the current contents of the list.
   lstCategories.get_Items().Clear();

   // Retrieve the categories.
   myCat2 = PerformanceCounterCategory.GetCategories();

   // Add the retrieved categories to the list.
   for (int i = 0; i < myCat2.length; i++)
```

```
    {
       lstCategories.get_Items().Add(myCat2[i].get_CategoryName());
    }
 }
```

To retrieve the counter list

1. In Design view, double-click the **Get Counters** button to access the Code Editor. The insertion point will be positioned in the **btnGetCounters_Click** event.
2. Add the following code to retrieve the list of counters from the selected category.

```
// Visual J#
private void btnGetCounters_Click (Object sender, System.EventArgs e)
{
   String[] instanceNames;
   System.Collections.ArrayList counters = new System.Collections.ArrayList();
   if (this.lstCategories.get_SelectedIndex() != -1)
   {
      PerformanceCounterCategory mycat = new PerformanceCounterCategory(lstCategories.get
_SelectedItem().ToString());

      // Remove the current contents of the list.
      lstCounters.get_Items().Clear();

      // Retrieve the counters.
      try
      {
         instanceNames = mycat.GetInstanceNames();
         if (instanceNames.length == 0)
         {
            counters.AddRange(mycat.GetCounters());
         }
         else
         {
            for (int i = 0; i < instanceNames.length; i++)
            {
               counters.AddRange(mycat.GetCounters(instanceNames[i]));
            }
         }

         // Add the retrieved counters to the list.
         int cnt = counters.get_Count();
         for (int i = 0; i < cnt; i++)
         {
      lstCounters.get_Items().Add(((PerformanceCounter) counters.get_Item(i)).get_Counter
Name());
         }
      }
      catch (System.Exception ex)
      {
         MessageBox.Show("Unable to list the counters for this category:\n" + ex.get_Mess
age());
      }
   }
}
```

To test your application

1. Save and compile your application.

2. Click the **Get Categories** button. You will see a list of categories in the list box.
3. Select one of the categories and click the **Get Counters** button. You will see a list of the counters for the selected category.

See Also

Monitoring Performance Thresholds | Introduction to Monitoring Performance Thresholds | Performance Counter Walkthroughs | Framework Services Component Walkthroughs

Walkthrough: Managing a Windows Process with Visual J#

The procedures in this topic walk you through the steps involved in creating a process, responding when a process stops, and stopping processes. In the first section, you will create a Windows application with **Button** controls to start and stop a **Notepad** process. You will start several instances of **Notepad** individually and then stop them as a group. In the second section, you will create a console application that enumerates the processes running on your computer.

For more information on using the **Process** component to interact with system processes, see Introduction to Monitoring and Managing Windows Processes.

To create your application

1. From the **New Project** dialog box, create a **Visual J# Windows Application**. Name the project **ProcessExample**.
2. With the Form1 designer open, click the **Windows Forms** tab in the Toolbox, and then add two buttons to the form.
3. In the Properties window, change the following properties:

Control	Property	Value
Button1	Name	**ButtonStart**
	Text	**Start Process**
Button2	Name	**ButtonStop**
	Text	**Stop Process**

4. Click the **Component** tab in the Toolbox, and then drag an instance of the **Process** component to the designer surface.
5. Name the component **myProcess**.

To start the Notepad.exe process

1. In the Properties window for the **myProcess** component instance, expand the **StartInfo** property, and set the **FileName** property to **notepad.exe**.
2. Double-click the **Start** button to access the Code Editor, and then add the following code to the **ButtonStart_Click** event:

```
// Visual J#
myProcess.Start();
```

3. Save all files, and then build and run the application.
4. Click the **Start Process** button a few times. You will see separate instances of Notepad for each click.
5. Close the individual **Notepad** applications.
6. Close the **Form1** application.

To close all current instances of the Notepad.exe process

1. Access the Code Editor for Form1.
2. Add the following lines of code at the beginning of the file just after the **package** statement:

```
// Visual J#
import System.Threading.*;
import System.Diagnostics.*;
```

3. From Design view, double-click the **Stop** button to access the **ButtonStop_Click** procedure.
4. Add the following code to close the current instances of Notepad.

```
// Visual J#
private void ButtonStop_Click (Object sender, System.EventArgs e)
{
   System.Diagnostics.Process[] myProcesses;
   myProcesses = myProcess.GetProcessesByName("Notepad");
   int count = myProcesses.length;
   for (int i = 0; i < count; i++)
   {
      System.Diagnostics.Process instance = myProcesses[i];
      instance.CloseMainWindow();
```

```
    }
}
```

5. Save all files, and then build and run your application.
6. Start several instances of Notepad by clicking the **Start Process** button.
7. Click the **Stop Process** button to immediately close all running instances of Notepad.
8. Close the **Form1** application.

To configure the component to wait for the Notepad process to finish running

In this procedure, you will freeze your code until the process finishes.

1. Access the Code Editor for Form1.
2. Add the following line of code to the **ButtonStop_Click** procedure before calling the **Process.CloseMainWindow** method:

```
// Visual J#
instance.WaitForExit(3000);
```

   Your **ButtonStop_Click()** procedure should now look like this:

```
// Visual J#
private void ButtonStop_Click (Object sender, System.EventArgs e)
{
   System.Diagnostics.Process[] myProcesses;
   myProcesses = myProcess.GetProcessesByName("Notepad");
   int count = myProcesses.length;
   for (int i = 0; i < count; i++)
   {
      System.Diagnostics.Process instance = myProcesses[i];
      instance.WaitForExit(3000);
      instance.CloseMainWindow();
   }
}
```

3. Save all files, and then build and run the application.
4. Start several instances of Notepad by clicking the **Start Processes** button.
5. Click the **Stop Processes** button to close the processes as before.

   You will notice that the application now waits 3 seconds for each process to stop before closing the next instance of the process.

6. Close the **Form1** application.

In this portion of the walkthrough, you will create a new console application that will retrieve and list the processes on the local machine.

To enumerate the processes on your computer

1. Create a new **Console Application** project.
2. Open the Code Editor for the module, and then add the following lines of code at the beginning of the file just after the **package** statement:

```
// Visual J#
import System.Threading.*;
import System.Diagnostics.*;
```

3. Add the following code to your module to enumerate the processes on your computer:

```
// Visual J#
public static void main(String[] args)
{
```

```
    System.Diagnostics.Process[] processes;
    System.Diagnostics.Process myProcess = null;
    processes = ((System.Diagnostics.Process)myProcess).GetProcesses();
    int count = processes.length;
    for (int n = 0; n < count; n++)
    {
       System.Diagnostics.Process instance = processes[n];
       System.Console.WriteLine(instance.get_ProcessName());
    }
    System.Threading.Thread.Sleep(5000);
 }
```

4. Save all files, and then build and run your application.

   The application will open a console window containing a list of all processes currently running on your machine. Calling **Thread.Sleep** pauses the console for 5 seconds before closing it.

   > **Note** You may receive exceptions in these two procedures if you call **GetProcesses** or **GetProcessesByName** and one of the returned processes finishes before you make the next call. In that case, you will receive an exception saying that the process does not exist.

See Also

Managing Processes | Starting Processes | Stopping Processes | Framework Services Component Walkthroughs

Walkthrough: Reacting to File System Events with Visual J#

The procedures in this topic walk you through the process of creating a **FileSystemWatcher** component, pointing it to a directory on your local computer, and then using the **Filter** property to watch only for changes to text files. You will create an event handler that responds when the **Created** and **Changed** events are raised, and you will use a form to display the notifications that result from these events.

Create the form and components you need for the application

In this procedure, you create a Windows Form and a **FileSystemWatcher** component instance that will work together to react to directory-level events.

1. From the **New Project** dialog box, create a **Visual J# Windows Application**, and name it **MyWatcher**.
2. From the **Windows Forms** tab in the Toolbox, drag two **Label** controls to the designer surface.
3. Click the **Components** tab in the Toolbox, and then drag the **FileSystemWatcher** icon to the designer surface for your component. By default, this component is named **fileSystemWatcher1**.

To set properties for your FileSystemWatcher component

In this procedure, you set several properties for your component to determine what it will watch. These settings cause the component to watch the specified directory on your local computer and to watch for the creation of files ending in the .txt file name extension.

1. Click the **FileSystemWatcher** component you created in the previous procedure, and view its properties in the Properties window.
2. Set the component's name to **myWatcher**.
3. Use the **Path** property to set the **FileSystemWatcher** component to watch a directory on your local computer. For example, on a computer running Microsoft® Windows® 2000, you might enter the following into the **Path** property to set the component to watch your My Documents directory:

   ```
   C:\Documents and Settings\yourusername\My Documents\
   ```

   Tip For the purposes of this example, you can use any directory you want on your local computer.

4. Set the **Filter** property to ***.txt** to have the component watch only those files ending in the .txt file name extension.

Write the necessary code for your component

In this procedure, you will define two event handlers for your component that define the processing that should occur whenever the **Changed** and **Created** events are raised.

1. Open the Code Editor for your form.
2. Add the following line at the top of the window to give you access to the **FileSystemWatcher** classes:

   ```
   // Visual J#
   import System.IO.*;
   ```

3. Return to Design view and double-click the **FileSystemWatcher** component. The Code Editor appears and a default event handler for the **Changed** event appears.

   Note For more information, see Event Handling in Windows Forms.

4. Use the following code to display a simple text string that will verify that your event was raised:

   ```
   // Visual J#
   private void myWatcher_Changed (Object sender, System.IO.FileSystemEventArgs e)
   {
      label1.set_Text("Changes made to: " + e.get_FullPath());
   }
   ```

5. Create the event handler for the **Created** event that specifies what your application should do whenever the component raises this event. For information on creating event handlers, see Creating Event Handlers on the Windows Forms Designer.

Add code as shown to display the full path of the newly created file.

```
// Visual J#
private void myWatcher_Created(Object sender, System.IO.FileSystemEventArgs e)
{
   label2.set_Text("The file: " + e.get_FullPath() + " has been added to your directory."
);
}
```

6. Save all files, and then build and run your application.

To test your FileSystemWatcher component

In this procedure, you will manually make changes in the directory your component is watching so that you can force the event handler to raise the **Changed** and **Created** events.

1. Run the application you created in the previous procedure.
2. Use Windows Explorer to find the directory that you set your **FileSystemWatcher** component to watch.
3. Open Notepad, and then create a new text file. Save this file to the directory you located in step 2 and close the file.

   Note This should raise the **Created** and **Changed** events and run the handler you defined.

4. Return to the form. You should see the created message in the label.
5. Open the text file you created, type a few lines of text, and save it again.

   Note This should raise the **Changed** event and run the handler you defined.

6. Return to the form. You should see the change message in the label.

See Also

Creating FileSystemWatcher Component Instances | Introduction to Monitoring File System Events | Framework Services Component Walkthroughs

Walkthrough: Storing and Retrieving Dynamic Properties with Visual J#

The procedures in this topic demonstrate the process of creating a project, storing some of its property settings in an external file, and then modifying the stored values and dynamically updating them in the deployed application.

In this application, you will create an instance of the Event Log component and store its property values to a file.

To create your component instance

1. On the **File** menu, point to **New**, and then click **Project**.
2. In the **New Project** dialog box, choose **Visual J# Projects** in the left pane, and then choose the **Windows Application** template. Name it **MyLog**.
3. From the **Components** tab of the Toolbox, drag an **EventLog** component to the form.

   Note Dynamic properties are not associated in any way with the **EventLog** component; you are simply using this component in this walkthrough as an example of the type of components whose properties you can store externally.

4. In the Properties window for the **EventLog** component, set the following values:

   - Set the **Log** property to **Application** so that your component can connect to a valid Windows event log.
   - Set the **MachineName** property to the name of the server on which the event log resides. Use a period (**.**) for the local machine.
   - Set the **Source** property to the string of your choice. In this case, you can use the name of your project.

5. On the **View** menu, click **Code** to open the Code Editor for Form1.
6. Expand the node titled **Windows Forms Designer generated code**.

   You should see a line showing the property value for the **Log** property, as follows:

```
// Visual J#
this.eventLog1.set_Log("Application");
```

7. Save your files.

   Note For more information on creating **EventLog** component instances and setting their properties, see Logging Application, Server, and Security Events.

To display the log property value

1. In the designer, drag a label and a button from the **Windows Forms** tab of the Toolbox to Form1.
2. Double-click the button to open the Code Editor and access the **Button** control's **Click** event handler.
3. Add this code to the method:

```
// Visual J#
label1.set_Text("You are connected to the " + eventLog1.get_Log() + " Log.");
```

To configure the Log property for your component

1. In the Properties window for the **EventLog** component, click the plus sign (+) next to **Dynamic Properties**.
2. Select the **Ellipsis** button next to the **Log** property.
3. In the **Dynamic Property** dialog box that appears, select the **Map property to a key in configuration file** check box, then click **OK**.

   This indicates that the **Log** property value should be stored using the default key — the class name and property name. In this case, the key will be **eventLog1.Log**.

4. Access the Code Editor for Form1, and find the **InitializeComponent** procedure for **MyLog**.

   You should see that code has automatically been changed to reflect that the value is now stored externally:

```
// Visual J#
this.eventLog1.set_Log(((String)(configurationAppSettings.GetValue(("eventLog1.Log"),
   String.class.ToType()))));
this.eventLog1.set_Source("MyLog");
```

5. Save your files and build the project.
6. In Solution Explorer, double-click the .config file for your project to view it in the editor. You should be able to locate the entry for the **EventLog1.Log** property value:

```
// Visual J#
<add key="eventLog1.Log" value="Application" />
```

7. Run your application. When you click the button, the label should say that you are connected to the Application log.
8. Close your application.

You also have the ability to modify the saved values outside of running the application. You might want to do this to set the initial values for an application.

To modify the stored value of the project .config file

1. In Notepad, open the .config file that is stored in the project folder.
2. Locate the **EventLog1.Log** entry you looked at in the previous procedure.
3. Change the value in that line to **System** or **Security**, and then save and close the file.
4. Press F5 to run the application. When you click the button, the new value should be reflected.

The configuration file in the project folder is named App.config. When you build the project, it is copied to the output folder and renamed to *Assemblyname*.config, where *assemblyname* is the project output, for example, MyLibrary.dll or MyApplication.exe. You need to modify this copy of the .config file to affect a deployed application. For more information, see Changing the Value of Dynamic Properties.

To modify the stored value of the deployed .config file

1. In Notepad, open the .config file for your application from its built location (often the \bin or \obj\debug folder of the project folder).
2. Locate the **EventLog1.Log** entry you looked at in the previous procedure.
3. Change the value in that line to **System** or **Security**, and then save and close the file.
4. Run the application from its built location. This is the same folder where you modified the .config file in step 1. When you click the button, the new value should be reflected.

See Also

Introduction to Dynamic Properties | Configuring Applications Using Dynamic Properties | Dynamic Properties Walkthrough

Walkthrough: Localizing Windows Forms with Visual J#

The Visual Studio project system provides considerable support for localizing Windows Forms applications. There are two ways to generate resource files using the Visual Studio development environment: one is to have the project system generate the resource files for localizable UI elements such as text and images on the form. The resource files are then built into satellite assemblies. The second way is to add a resource file template and then edit the template with the XML Designer. A reason for doing the latter is to make localizable strings that appear in dialog boxes and error messages. You must then write code to access these resources.

This walkthrough topic demonstrates both processes in a single Windows Application project.

You can also convert a text file to a resource file; for more information, see Resources in Text File Format and Resource File Generator (Resgen.exe).

To have Visual Studio generate resource files for you

1. Create a new Windows Application named "WindowsApplication1". For details see Creating a Windows Application Project.
2. In the Properties window, set the form's **Localizable** property to **true**.

   The **Language** property already is set to **(Default)**.
3. Drag a **Button** control from the Windows Forms tab of the **Toolbox** to the form, and set its **Text** property to "Hello World".
4. Set the form's **Language** property to "German (Germany)".
5. Set the button's **Text** property to "Hallo Welt".
6. Set the form's **Language** property to "French (France)".
7. Set the button's **Text** property to "Bonjour le Monde". You can resize the button to accommodate the longer string, if necessary.
8. Save and build the solution.
9. Click the "Show All Files" button in Solution Explorer.

   The resource files appear underneath Form1.jsl. Form1.resx is the resource file for the fallback culture, which will be built into the main assembly. Form1.de-DE.resx is the resource file for German as spoken in Germany. Form1.fr-FR.resx is the resource file for French as spoken in France.

   In addition, you will see files appear named Form1.de.resx and Form1.fr.resx. Visual Studio automatically creates these files in order to work around a limitation in Visual SourceSafe having to do with adding new files to a project during a save operation. The .resx files are empty and contain no resources.
10. Press the F5 key or choose **Start** from the **Debug** menu.

    You will now see a dialog box with an English, French, or German greeting depending on the UI language of your operating system.

The following procedure shows you how to set the UI culture so the application displays your French resources. In real-world applications, you would not hard-code the UI culture in this way. The setting for the UI culture would depend instead on a user setting or an application setting.

To set the UI culture to view specific resources

1. In the Code Editor, add the following code at the beginning of the module, before the Form1 declaration:

```
// Visual J#
import System.Globalization.*;
import System.Threading.*;
```

2. Add the following code. In Visual J#, it should go in **Form1**() and also before calling the **InitializeComponent** function.

```
// Visual J#
// Sets the UI culture to French (France)
System.Threading.Thread.get_CurrentThread().set_CurrentUICulture( new CultureInfo("fr-FR"
));
```

3. Save and build the solution.

4. Press the F5 key or choose **Start** from the **Debug** menu.

   Now the form will be always displayed in French. If you changed the size of the button earlier to accommodate the longer French string, notice that the button size has also been persisted in the French resource file.

To manually add resource files to the project and edit them

1. On the **Project** menu, click **Add New Item**.
2. In the Templates box, select the **Assembly Resource File** template. Type the file name "WinFormStrings.resx" in the **Name** box. The file WinFormStrings.resx will contain fallback resources in English. These resources will be accessed whenever the application cannot find resources more appropriate to the UI culture.

   The file is added to your project in Solution Explorer and automatically opens in the XML Designer in Data view.

3. In the **Data Tables** pane, select **data**.
4. In the **Data** pane, click an empty row and enter "strMessage" in the **name** column and "Hello World" in the **value** column.

   You do not need to specify the **type** or **mimetype** for a string; they are used for objects. The type specifier holds the data type of the object being saved. The MIME type specifier holds the base type (base64) of the binary information stored, if the object consists of binary data.

5. On the **File** menu, click **Save WinFormStrings.resx**.
6. Do steps 1-5 twice more to create two more resource files named "WinFormStrings.de-DE.resx" and "WinFormStrings.fr-FR.resx", with the string resources specified in the following table. The file WinFormStrings.de-DE.resx will contain resources that are specific to German as spoken in Germany. The file WinFormStrings.fr-FR.resx will contain resources that are specific to French as spoken in France.

Resource file name	Name	Value
WinFormStrings.de-DE.resx	strMessage	Hallo Welt
WinFormStrings.fr-FR.resx	strMessage	Bonjour le Monde

To access the manually added resources

1. In the Code Editor, import the `System.Resources` namespace at the beginning of the code module.

```
// Visual J#
import System.Resources.*;
```

2. In Design view, double-click the button to display the code for its **Click** event handler and add the following code. The **ResourceManager** constructor takes two arguments. The first is the root name of the resources — that is, the name of the resource file without the culture and .resx suffixes. The second argument is the main assembly.

   In this walkthrough, no namespaces are declared, so the first argument of the **ResourceManager** constructor can take the form of "ProjectName.ResourceFileRootName". However, in a real-world application, you would set the DefaultNamespace property. In that case, you would need to declare the resource manager with the fully qualified root name of the resource file, including its namespace. For example, if the default namespace is "MyCompany.MyApplication.MyComponent", the first argument of the **ResourceManager** constructor might be "MyCompany.MyApplication.MyComponent.WinFormStrings".

```
// Visual J#
// Declare a Resource Manager instance
ResourceManager LocRM = new ResourceManager("WindowsApplication1.WinFormStrings",
   System.Type. GetType("WindowsApplication1.Form1").get_Assembly());
// Assign the string for the "strMessage" key to a messagebox
MessageBox.Show(LocRM.GetString("strMessage"));
```

   Note By default, the **ResourceManager** object is case-sensitive. If you wish to do case-insensitive lookups so that "TXTWELCOME" retrieves the same resource as "txtWelcome", you can set the resource manager's IgnoreCase property to **true**. However, for performance reasons, it is best to always specify the correct case for your resource names. Doing case-insensitive resource lookups can cause performance problems.

3. Build and run the form. Click the button.

   The message box will display a string appropriate for the UI culture setting; or if it cannot find a resource for the UI culture, it

will display a string from the fallback resources.

See Also

Setting the Culture and UI Culture for Windows Forms Globalization | Introduction to International Applications in Visual Basic and Visual C# | Walkthrough: Localizing Web Forms Pages with Visual J# | Globalizing and Localizing Applications | Localization Walkthroughs

Walkthrough: Localizing Web Forms Pages with Visual J#

When you localize Web Forms pages, the Visual Studio project system does not automatically generate resource files. You must manually create and edit the XML resource files. This walkthrough topic demonstrates how to add a resource file template and then edit the template with the XML Designer to create resources in English, German, and German as spoken in Austria. This topic also discusses how to write code to access these resources.

You can also convert a text file to a resource file. For more information, see Resources in Text File Format and Resource File Generator (Resgen.exe).

To manually create and edit a resource file

1. Create a new ASP.NET Web Application named "LocProject". For details, see Creating Web Projects.
2. On the **Project** menu, click **Add New Item**.
3. In the Templates box, select the **Assembly Resource File** template. Type the file name "strings.resx" in the **Name** box. The file strings.resx will contain fallback resources in English. These resources will be accessed whenever the application cannot find resources more appropriate to the UI culture.

   The file is added to your project in Solution Explorer and automatically opens in the XML Designer in Data view.

4. In the **Data Tables** pane, select **data**.
5. In the **Data** pane, click an empty row and enter "txtSearch" in the **name** column and "Search for the following text:" in the **value** column.

   You do not need to specify the **type** or **mimetype** for a string; they are used for objects. The type specifier holds the data type of the object being saved. The MIME type specifier holds the base type (base64) of the binary information stored, if the object consists of binary data.

6. Click another empty row and enter "txtWelcome" in the **name** column and "Welcome to Localization!" in the **value** column.
7. On the File menu, click **Save strings.resx**.
8. Do steps 2-7 twice more to create two more resource files named "strings.de.resx" and "strings.de-AT.resx", containing the string resources specified in the following table. The file strings.de.resx will contain German resources that are neutral — that is, they apply to any region where German is spoken. The file strings.de-AT.resx will contain German resources that are specific to Austria.

Resource file name	Name	Value
strings.de.resx	txtSearch	Nach dem folgenden Text suchen:
strings.de.resx	txtWelcome	Willkommen zur Lokalisierung!
strings.de-AT.resx	txtWelcome	Grüßt's euch zur ASP.NET Lokalisierung !

   You can enter the accented characters by copying and pasting from this help topic, using a German keyboard layout, or by using the character map application charmap.exe.

To access the resources

1. In the Code Editor, import the `System.Resources` and `System.Globalization` namespaces at the beginning of the code module. This code should occur after the **package** declaration.

```
// Visual J#
import System.Resources.*;
import System.Globalization.*;
```

2. Add the following code to the declarations at the top of the WebForm1 class.

```
// Visual J#
protected ResourceManager LocRM;
```

3. Add the following code to the **Page_Load** method in the WebForm1 code-behind class. The **ResourceManager** constructor takes two arguments. The first is the root name of the resources — that is, the name of the resource file without the culture and .resx suffixes. The second argument is the main assembly.

```
// Visual J#
```

```
LocRM= new ResourceManager("LocProject.strings", System.Type.GetType("LocProject.WebForm1
").get_Assembly());
```

To display a static resource string in the page

1. In HTML view, add some static text from a resource file with the following HTML code. Place this code between the <form> and </form> tags.

```
<h3><%=LocRM.GetString("txtWelcome")%></h3>
```

> **Note** By default, the **ResourceManager** object is case-sensitive. If you wish to do case-insensitive lookups so that "TXTWELCOME" retrieves the same resource as "txtWelcome", you can set the resource manager's IgnoreCase property to true. However, for performance reasons, it is best to always specify the correct case for your resource names. Doing case-insensitive resource lookups can cause performance problems.

To assign a resource string to the property of a control

1. In Design view, drag a **Button** control from the Web Forms tab of the Toolbox to your form.
2. In the Properties window, delete the value for the button's **Text** property
3. In the Code Editor, add the following code to the **Page_Load** method to retrieve the text for the button from the resource file.

```
// Visual J#
Button1.set_Text(LocRM.GetString("txtSearch"));
```

4. In the Code Editor, add the following code to import the `System.Threading` namespace at the beginning of the code module.

```
// Visual J#
// Import this namespace at the beginning of the code module.
import System.Threading.*;
```

5. Add the following code to the very beginning of the **Page_Load** method to specify that the Web Forms page should display resources and formatting according to the culture specified in the browser's Accept-Language header. This code is important because otherwise your Web application will default to the culture settings of the Web server, which may not be appropriate for the remote user viewing the Web Forms page.

```
// Visual J#
// Set the culture and UI culture to the browser's accept language
System.String[] lang =this.get_Request().get_UserLanguages();
System.Threading.Thread.get_CurrentThread().set_CurrentCulture(CultureInfo.CreateSpecific
Culture(lang[0]));
System.Threading.Thread.get_CurrentThread().set_CurrentUICulture(new CultureInfo(lang[0])
);
```

6. Save the project.
7. On the **Build** menu, click the **Build LocProject** command.
8. Right-click the WebForm1.aspx file in Solution Explorer and choose **View in Browser**. If you are running on a German or Austrian operating system you will see the page in German or Austrian; otherwise you will see it in English.

   If you view the page first on a system configured for German and then on a system configured for Austrian, you will notice that the Welcome text has changed to the Austrian resources, but the text for the search button has not changed. This is an example of the resource fallback system: the Austrian resources only changed the welcome text, and the remaining text defaults to the neutral German resource.

See Also

Setting the Culture and UI Culture for Web Forms Globalization | Introduction to International Applications in Visual Basic and Visual C# | Walkthrough: Localizing Windows Forms with Visual J# | Globalizing and Localizing Applications | Localization Walkthroughs

Expressions in Visual J#

The Visual J# debugger includes an expression evaluator that evaluates expressions entered in the QuickWatch dialog box, in the Watch window, or in the Autos, Local, or This/Me windows. The expression evaluator is also at work in the Breakpoints window. If you type the name of a function where you want to set a breakpoint, for example, the expression evaluator handles it.

The following topics offer specific information and discuss some of the expression types that are not supported:

- Identifiers and Types
- Function Evaluation
- Operators
- Arrays
- Strings
- Casts
- Object Comparison and Assignment
- Property Evaluation
- WebMethods

The debugger uses auto-expand rules to display the contents of a data type in meaningful form. If you need to, you can add custom auto-expand elements to display your own custom data types. For more information, see Displaying Elements of a Custom Data Type.

Identifiers and Types

Debugger expressions can use any identifier visible within the current scope. If the debugger is halted in function `Test`, for example, you can use any identifier visible within `Test`, including constants, variable names, and method names.

The debugger can correctly display any variable of a primitive or intrinsic type. For variables of class type, the debugger correctly displays the value based on the most-derived type. If you have an object `leo` of type `lion`, derived from type `cat`, you can evaluate `leo.clawlength` and get the correct value for an object of type `lion`.

In Visual J#, you can assign a new value to any left-hand expression that is an l-value and not an array. This includes **primitive**, **class** and **Object** types.

Function Evaluation

The debugger supports the evaluation of functions, including overloaded functions. Therefore, you can enter either of the following expressions, and the debugger will call the correct version of the overloaded function:

```
kangaroo ()
kangaroo (joey)
```

Evaluating a function in the debugger actually calls and executes the code for that function. If the function has side effects, such as allocating memory or changing the value of a global variable, evaluating the function in a debugger window will change the state of your program, which can produce unexpected results.

When you set a breakpoint on an overloaded function, the location of the breakpoint depends on how you specify the function. If you specify only the function name, the debugger will set one breakpoint on each overload of that function name. If you specify the complete signature (function name and full argument list), the debugger sets one breakpoint on the specified overload.

Function evaluation on the following **java.lang.Object** methods is not supported:

- **wait**
- **notify**
- **notifyAll**
- **getClass**

Operators

The debugger correctly evaluates most built-in operators, including:

- Relational operators. Examples: (*expr1* > *expr2*, *expr1* < *expr2*, *expr1* <= *expr2*, *expr1* >= *expr2*, *expr1* == *expr2*, *expr1* !=

expr2).

- Boolean operators. Examples: (*expr1* && *expr2*, *expr1* || *expr2*).
- Conditional operator. Example: (*expr1* ? *expr2* : *expr3*).
- Arithmetical operators. Examples: (*expr1* + *expr2*, *expr1* - *expr2*, *expr1* * *expr2*, *expr1* / *expr2*, *expr1* % *expr2*).
- Bitwise operators. Examples: (*expr1* & *expr2*, *expr1* ^ *expr2*, *expr1* | *expr2*).
- Shift operators. Examples: (*expr1* >> *expr2*, *expr1* << *expr2*, *expr1* >>> *expr2*).
- Assignment operators. Examples: (*lvalue* = *expr2*, *lvalue* *= *expr2*, *lvalue* /= *expr2*, *lvalue* %= *expr2*, *lvalue* += *expr2*, *lvalue* -= *expr2*, *lvalue* <<= *expr2*, *lvalue* >>= *expr2*, *lvalue* &= *expr2*, *lvalue* ^= *expr2*, *lvalue* |= *expr2*, *lvalue* >>>= *expr2*).
- Unary operators. Examples: (+*expr1*, - *expr1*, *expr1*++, ++*expr1*, *expr1*--, --*expr1*, ~*expr1*).

You can use the comma operator to enter a series of expressions: *expr1, expr2*, *expr3*.

Arrays

Visual J# has two forms of arrays: ranked arrays and jagged arrays.

A ranked array uses a separate [] operator for each dimension:

```
X[ expr1 ][ expr2 ]
```

A jagged array uses only one [] operator and uses commas to separate the dimensions:

```
X[ expr1, expr2 ]
```

Strings

Strings typed into the debugger windows within quotes are equivalent to the **System.String** type. The debugger recognizes the indexed operator when it is used with strings as well as arrays. So, for example, you can enter:

```
"hello world"[0]
```

The Watch window will display the correct value:

```
'h'
```

In Visual J#, unlike native C/C++, you can edit the value of a string in the debugger. In addition, you can use the `Length` operator on a string:

```
mystring.Length
```

-or-

```
"hello world".Length
```

In Visual J#, you can concatenate **String** objects:

```
"hello" + " world"
```

In Visual J#, you can also evaluate methods on strings. For example, if the source contains `String str = "Hello World";`, you can evaluate the **equals** method in the Watch window by typing the following expression:

```
str.equals( "Hello World")
```

This expression returns true in this example.

Casts

Simple cast expressions work in the debugger:

```
(A)x
```

User-defined casts work in the debugger.

Object Comparison and Assignment

Expressions that compare or assign Visual J# objects work in the debugger:

```
obj1 == obj2
obj1 = obj2
```

Property Evaluation

The debugger can evaluate properties in any variable window. However, evaluating a property in the debugger can have side effects that produce unexpected and undesired results. To protect against side effects caused by accidental evaluation, you can turn property evaluation off in the Options dialog box.

Properties can also be evaluated by their **get_** and **set_** methods. For example, if an object has a property `text`, then `obj.get_text()` will show the value of the property in the Watch window.

WebMethods

You cannot call WebMethods from debugger windows.

See Also

Debugging in Visual J# | Expressions in the Debugger | QuickWatch Dialog Box | Autos Window | Locals Window | Breakpoints Window

Compiler Error VJS1030

'*file*' is a binary file and not a source file

A binary file, such as a .dll, was passed to the compiler as a source code file.

Compiler Error VJS1274

Cannot author attributes, enums or value types

Value types, enums, and attributes cannot be extended.

The following sample generates VJS1274:

```
// VJS1274a.jsl
class Class1 extends System.ValueType   // VJS1274
{
}
```

```
// VJS1274b.jsl
class Class2 extends System.Enum   // VJS1274
{
}
```

```
// VJS1274c.jsl
class Class3 extends System.Attribute   // VJS1274
{
}
```

Compiler Warning (level 1) VJS1466

The attribute System.CLSCompliantAttribute is being attached but the compiler will not verify that the assembly is CLS compliant

CLSCompliantAttribute was used and ignored by the Visual J# compiler, which does not validate for compliance with the Common Language Specification (CLS).

The following sample generates VJS1466:

```
// VJS1466.jsl
// compile with: /W:1
/** @attribute System.CLSCompliant (true) */   // VJS1466
public class MyClass
{
   public static void main (String [] args)
   {
   }
}
```

Compiler Warning (level 1) VJS1501

Method '*method*' is not compatible with base class method '*base method*'. The incompatibility is being ignored

This warning is given when the compiler detects an override of a method in **System.Object** but with a different signature. This may happen with legacy Java language code.

The following sample generates VJS1501:

```
// VJS1501.jsl
// compile with: /W:1 /target:library
class MyClass
{
   public int GetType()   // VJS1501, GetType is in System.Object but signature differs
   // try the following line instead
   // public int GetMyType()
   {
      return 0;
   }
}
```

Compiler Error VJS1502

'Package' scope is being used across assemblies with secure scoping option. Might throw an exception for invalid access at runtime.

The /securescoping compiler option was used on an assembly, which then was referenced and accessed by another assembly using the same package. If the caller was not in the same package, the call would not be valid.

The following sample generates VJS1502:

```
// VJS1502a.jsl
// compile with: /securescoping /target:library
package p;
public class MyClass
{
   void Test()
   {
   }   // package-scoped member
}
```

```
// VJS1502b.jsl
// compile with: /reference:VJS1502a.dll /target:library
package p;
public class b
{
   public void Test2(MyClass A)
   {
      A.Test();   // VJS1502 using package access
   }
}
```

Compiler Error VJS2106

Identifier expected, '*token*' is a keyword

A reserved keyword was detected where an identifier was expected.

The following sample generates VJS2106:

```
public class MyClass
{
   int int;
}
```

Issues in Proxy Generation for Web Services

In a Visual J# project, when you add a reference to an XML Web service, a proxy for the Web service is generated and added to the project. This proxy is used to invoke the Web services method. The CodeDOM provider generates the source code for the proxy. There are certain cases when proxy generation may fail due to language features that are not available in Visual J#, as follows:

- The Web services method has arguments that are passed by reference.
- The parameters are in/out parameters.
- The parameters or return types are enumerations.

In such cases, the CodeDOM provider inserts error statements into the generated source file, so that there is a compile-time error in your project.

An alternative is to generate the proxy in Visual Basic or Visual C# and use it in the Visual J# project in the following ways:

1. Add a Visual Basic or Visual C# control library project to your solution.
2. Add the Web reference to the new project.
3. Add this project as a reference to your Visual J# project.

-or-

1. Open a Visual Studio .NET command prompt.
2. Use the wsdl.exe tool to generate the proxy class for the Web reference in Visual Basic or Visual C#.
3. Compile the generated file to a library (.dll) by using appropriate command line compiler for the language (csc.exe or vbc.exe).
4. Add the DLL to the Visual J# project as a reference.

   Note For Web service methods with enumerations, Visual J# can generate proxies successfully if the document/literal style is used. That is, in the WSDL service definition, if the **style** attribute of the <*soap*:*binding*> element is set to **document** (and not **rpc**) and the **use** attribute of the <*soap*:*body*> binding element is not set to **Encoded**, SOAP messages that communicate with the Web service are not encoded. In this scenario, a proxy class with constant values and a string field is generated for the enumeration type. However, if SOAP messages are encoded, proxy class generation for such a Web services method is not supported in Visual J#.

See Also

Language Support | Generating and Compiling Source Code Dynamically in the Java Language

.NET Framework Class Library (Visual J# Syntax)

The topics in this section show Visual J# syntax for the .NET Framework class library. Each topic contains links to corresponding .NET Framework topics where you can get details on parameters, return values, and usage. This section is organized by namespace. It also includes the topic Accessing .NET Framework SDK Reference Topics from Visual J#.

Microsoft.CSharp Namespace (Visual J# Syntax)

Microsoft.JScript Namespace (Visual J# Syntax)

Microsoft.VisualBasic Namespace (Visual J# Syntax)

Microsoft.Vsa Namespace (Visual J# Syntax)

Microsoft.Win32 Namespace (Visual J# Syntax)

System Namespace (Visual J# Syntax)

System.CodeDom Namespace (Visual J# Syntax)

System.CodeDom.Compiler Namespace (Visual J# Syntax)

System.Collections Namespace (Visual J# Syntax)

System.Collections.Specialized Namespace (Visual J# Syntax)

System.ComponentModel Namespace (Visual J# Syntax)

System.ComponentModel.Design Namespace (Visual J# Syntax)

System.ComponentModel.Design.Serialization Namespace (Visual J# Syntax)

System.Configuration Namespace (Visual J# Syntax)

System.Configuration.Assemblies Namespace (Visual J# Syntax)

System.Configuration.Install Namespace (Visual J# Syntax)

System.Data Namespace (Visual J# Syntax)

System.Data.Common Namespace (Visual J# Syntax)

System.Data.OleDb Namespace (Visual J# Syntax)

System.Data.SqlClient Namespace (Visual J# Syntax)

System.Data.SqlTypes Namespace (Visual J# Syntax)

System.Diagnostics Namespace (Visual J# Syntax)

System.Diagnostics.SymbolStore Namespace (Visual J# Syntax)

System.DirectoryServices Namespace (Visual J# Syntax)

System.Drawing Namespace (Visual J# Syntax)

System.Drawing.Design Namespace (Visual J# Syntax)

System.Drawing.Drawing2D Namespace (Visual J# Syntax)

System.Drawing.Imaging Namespace (Visual J# Syntax)

System.Drawing.Printing Namespace (Visual J# Syntax)

System.Drawing.Text Namespace (Visual J# Syntax)

System.EnterpriseServices Namespace (Visual J# Syntax)

System.EnterpriseServices.CompensatingResourceManager Namespace (Visual J# Syntax)

System.Globalization Namespace (Visual J# Syntax)

System.IO Namespace (Visual J# Syntax)

System.IO.IsolatedStorage Namespace (Visual J# Syntax)

System.Management Namespace (Visual J# Syntax)

System.Management.Instrumentation Namespace (Visual J# Syntax)

System.Messaging Namespace (Visual J# Syntax)

System.Net Namespace (Visual J# Syntax)

System.Net.Sockets Namespace (Visual J# Syntax)

System.Reflection Namespace (Visual J# Syntax)

System.Reflection.Emit Namespace (Visual J# Syntax)

System.Resources Namespace (Visual J# Syntax)

System.Runtime.CompilerServices Namespace (Visual J# Syntax)

System.Runtime.InteropServices Namespace (Visual J# Syntax)

System.Runtime.InteropServices.CustomMarshalers Namespace (Visual J# Syntax)

System.Runtime.InteropServices.Expando Namespace (Visual J# Syntax)

System.Runtime.Remoting Namespace (Visual J# Syntax)

System.Runtime.Remoting.Activation Namespace (Visual J# Syntax)

System.Runtime.Remoting.Channels Namespace (Visual J# Syntax)

System.Runtime.Remoting.Channels.Http Namespace (Visual J# Syntax)

System.Runtime.Remoting.Channels.Tcp Namespace (Visual J# Syntax)

System.Runtime.Remoting.Contexts Namespace (Visual J# Syntax)

System.Runtime.Remoting.Lifetime Namespace (Visual J# Syntax)

System.Runtime.Remoting.Messaging Namespace (Visual J# Syntax)

System.Runtime.Remoting.Metadata Namespace (Visual J# Syntax)

System.Runtime.Remoting.Metadata.W3cXsd2001 Namespace (Visual J# Syntax)

System.Runtime.Remoting.MetadataServices Namespace (Visual J# Syntax)

System.Runtime.Remoting.Proxies Namespace (Visual J# Syntax)

System.Runtime.Remoting.Services Namespace (Visual J# Syntax)

System.Runtime.Serialization Namespace (Visual J# Syntax)

System.Runtime.Serialization.Formatters Namespace (Visual J# Syntax)

System.Runtime.Serialization.Formatters.Binary Namespace (Visual J# Syntax)

System.Runtime.Serialization.Formatters.Soap Namespace (Visual J# Syntax)

System.Security Namespace (Visual J# Syntax)

System.Security.Cryptography Namespace (Visual J# Syntax)

System.Security.Cryptography.X509Certificates Namespace (Visual J# Syntax)

System.Security.Cryptography.Xml Namespace (Visual J# Syntax)

System.Security.Permissions Namespace (Visual J# Syntax)

System.Security.Policy Namespace (Visual J# Syntax)

System.Security.Principal Namespace (Visual J# Syntax)

System.ServiceProcess Namespace (Visual J# Syntax)

System.Text Namespace (Visual J# Syntax)

System.Text.RegularExpressions Namespace (Visual J# Syntax)

System.Threading Namespace (Visual J# Syntax)

System.Timers Namespace (Visual J# Syntax)

System.Web Namespace (Visual J# Syntax)

System.Web.Caching Namespace (Visual J# Syntax)

System.Web.Configuration Namespace (Visual J# Syntax)

System.Web.Hosting Namespace (Visual J# Syntax)

System.Web.Mail Namespace (Visual J# Syntax)

System.Web.Security Namespace (Visual J# Syntax)

System.Web.Services Namespace (Visual J# Syntax)

System.Web.Services.Configuration Namespace (Visual J# Syntax)

System.Web.Services.Description Namespace (Visual J# Syntax)

System.Web.Services.Discovery Namespace (Visual J# Syntax)

System.Web.Services.Protocols Namespace (Visual J# Syntax)

System.Web.SessionState Namespace (Visual J# Syntax)

System.Web.UI Namespace (Visual J# Syntax)

System.Web.UI.Design Namespace (Visual J# Syntax)

System.Web.UI.Design.WebControls Namespace (Visual J# Syntax)

System.Web.UI.HtmlControls Namespace (Visual J# Syntax)

System.Web.UI.WebControls Namespace (Visual J# Syntax)

System.Windows.Forms Namespace (Visual J# Syntax)

System.Windows.Forms.Design Namespace (Visual J# Syntax)

System.Xml Namespace (Visual J# Syntax)

System.Xml.Schema Namespace (Visual J# Syntax)

System.Xml.Serialization Namespace (Visual J# Syntax)

System.Xml.XPath Namespace (Visual J# Syntax)

System.Xml.Xsl Namespace (Visual J# Syntax)

Accessing .NET Framework SDK Reference Topics from Visual J#

When working in the Code Editor, you will not see .NET Framework SDK reference topics displayed in the Dynamic Help window.

To access help on .NET Framework elements in Visual J# files in the Code Editor, place the cursor in the .NET Framework code for which you would like help and press F1. The Index window appears, and you can navigate to the proper topic. In the See Also section of the .NET Framework reference topic, a reciprocal link to the Visual J# documentation appears.

See Also

Introducing Visual J#

Microsoft.CSharp Namespace (Visual J# Syntax)

Classes

Compiler

CompilerError

CSharpCodeProvider

Enumerations

ErrorLevel

Compiler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Compiler
   extends System.Object
```

Public Constructors

```
public Compiler();
```

Public Methods

```
public static Microsoft.CSharp.CompilerError[] Compile(
   System.String[] sourceTexts,
   System.String[] sourceTextNames,
   System.String target,
   System.String[] imports,
   System.Collections.IDictionary options
);
```

CompilerError Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CompilerError
   extends System.Object
```

Public Constructors

```
public CompilerError();
```

Public Methods

```
public System.String ToString();
```

Public Fields

```
public Microsoft.CSharp.ErrorLevel ErrorLevel;
public System.String ErrorMessage;
public int ErrorNumber;
public int SourceColumn;
public System.String SourceFile;
public int SourceLine;
```

CSharpCodeProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CSharpCodeProvider
   extends System.CodeDom.Compiler.CodeDomProvider
```

Public Constructors

```
public CSharpCodeProvider();
```

Public Properties

```
public System.String get_FileExtension();
```

Public Methods

```
public System.CodeDom.Compiler.ICodeCompiler CreateCompiler();
public System.CodeDom.Compiler.ICodeGenerator CreateGenerator();
public System.ComponentModel.TypeConverter GetConverter(
   System.Type type
);
```

ErrorLevel Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ErrorLevel
```

Members

Error

FatalError

None

Warning

Microsoft.JScript Namespace (Visual J# Syntax)

Classes

JScriptCodeProvider

JScriptCodeProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerCategoryAttribute(Category = "code") */
public class JScriptCodeProvider
   extends System.CodeDom.Compiler.CodeDomProvider
```

Public Constructors

```
public JScriptCodeProvider();
```

Public Properties

```
public System.String get_FileExtension();
```

Public Methods

```
public System.CodeDom.Compiler.ICodeCompiler CreateCompiler();
public System.CodeDom.Compiler.ICodeGenerator CreateGenerator();
```

Microsoft.VisualBasic Namespace (Visual J# Syntax)

Classes

VBCodeProvider

VBCodeProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class VBCodeProvider
   extends System.CodeDom.Compiler.CodeDomProvider
```

Public Constructors

```
public VBCodeProvider();
```

Public Properties

```
public System.String get_FileExtension();
public System.CodeDom.Compiler.LanguageOptions get_LanguageOptions();
```

Public Methods

```
public System.CodeDom.Compiler.ICodeCompiler CreateCompiler();
public System.CodeDom.Compiler.ICodeGenerator CreateGenerator();
public System.ComponentModel.TypeConverter GetConverter(
   System.Type type
);
```

Microsoft.Vsa Namespace (Visual J# Syntax)

Classes

VsaException

VsaLoader

VsaModule

Interfaces

IVsaCodeItem

IVsaDTCodeItem

IVsaDTEngine

IVsaEngine

IVsaError

IVsaGlobalItem

IVsaIDE

IVsaIDESite

IVsaItem

IVsaItems

IVsaPersistSite

IVsaReferenceItem

IVsaSite

Enumerations

VsaError

VsaIDEMode

VsaItemFlag

VsaItemType

IVsaCodeItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "E0C0FFE7-7eea-4ee5-b7e4-
0080c7eb0b74") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface IVsaCodeItem
   implements Microsoft.Vsa.IVsaItem
```

Public Properties

```
System.CodeDom.CodeObject get_CodeDOM();
System.String get_SourceText();
void set_SourceText(
   System.String value
);
```

Public Methods

```
void AddEventSource(
   System.String eventSourceName,
   System.String eventSourceType
);
void AppendSourceText(
   System.String text
);
void RemoveEventSource(
   System.String eventSourceName
);
```

IVsaDTCodeItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "E0C0FFED-7eea-4ee5-b7e4-
0080c7eb0b74") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface IVsaDTCodeItem
```

Public Properties

```
boolean get_CanDelete();
void set_CanDelete(
   boolean value
);
boolean get_CanMove();
void set_CanMove(
   boolean value
);
boolean get_CanRename();
void set_CanRename(
   boolean value
);
boolean get_Hidden();
void set_Hidden(
   boolean value
);
boolean get_ReadOnly();
void set_ReadOnly(
   boolean value
);
```

IVsaDTEngine Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "E0C0FFEE-7eea-4ee5-b7e4-
0080c7eb0b74") */
public interface IVsaDTEngine
```

Public Properties

```
System.String get_TargetURL();
void set_TargetURL(
   System.String value
);
```

Public Methods

```
void AttachDebugger(
   boolean isAttach
);
Microsoft.Vsa.IVsaIDE GetIDE();
void InitCompleted();
```

IVsaEngine Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "E0C0FFE1-7eea-4ee2-b7e4-
0080c7eb0b74") */
public interface IVsaEngine
```

Public Properties

```
System.Reflection.Assembly get_Assembly();
System.Security.Policy.Evidence get_Evidence();
void set_Evidence(
   System.Security.Policy.Evidence value
);
boolean get_GenerateDebugInfo();
void set_GenerateDebugInfo(
   boolean value
);
boolean get_IsCompiled();
boolean get_IsDirty();
boolean get_IsRunning();
Microsoft.Vsa.IVsaItems get_Items();
System.String get_Language();
int get_LCID();
void set_LCID(
   int value
);
System.String get_Name();
void set_Name(
   System.String value
);
System.String get_RootMoniker();
void set_RootMoniker(
   System.String value
);
System.String get_RootNamespace();
void set_RootNamespace(
   System.String value
);
Microsoft.Vsa.IVsaSite get_Site();
void set_Site(
   Microsoft.Vsa.IVsaSite value
);
System.String get_Version();
```

Public Methods

```
void Close();
boolean Compile();
System.Object GetOption(
   System.String name
);
void InitNew();
boolean IsValidIdentifier(
   System.String identifier
);
```

```
void LoadSourceState(
   Microsoft.Vsa.IVsaPersistSite site
);
void Reset();
void RevokeCache();
void Run();
void SaveCompiledState(
   System.Byte[]& pe,
   System.Byte[]& pdb
);
void SaveSourceState(
   Microsoft.Vsa.IVsaPersistSite site
);
void SetOption(
   System.String name,
   System.Object value
);
```

IVsaError Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "E0C0FFE4-7eea-4ee2-b7e4-
0080c7eb0b74") */
public interface IVsaError
```

Public Properties

```
System.String get_Description();
int get_EndColumn();
int get_Line();
System.String get_LineText();
int get_Number();
int get_Severity();
Microsoft.Vsa.IVsaItem get_SourceItem();
System.String get_SourceMoniker();
int get_StartColumn();
```

IVsaGlobalItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "4E76D92E-E29D-46f3-AE22-
0333158109F1") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface IVsaGlobalItem
   implements Microsoft.Vsa.IVsaItem
```

Public Properties

```
boolean get_ExposeMembers();
void set_ExposeMembers(
   boolean value
);
void set_TypeString(
   System.String value
);
```

IVsaIDE Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "78470A10-8153-407d-AB1B-
05067C54C36B") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface IVsaIDE
```

Public Properties

```
System.String get_DefaultSearchPath();
void set_DefaultSearchPath(
   System.String value
);
System.Object get_ExtensibilityObject();
Microsoft.Vsa.VsaIDEMode get_IDEMode();
Microsoft.Vsa.IVsaIDESite get_Site();
void set_Site(
   Microsoft.Vsa.IVsaIDESite value
);
```

Public Methods

```
void EnableMainWindow(
   boolean isEnable
);
void ShowIDE(
   boolean showOrHide
);
```

IVsaIDESite Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "7BD84086-1FB5-4b5d-8E05-
EAA2F17218E0") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface IVsaIDESite
```

Public Methods

```
void Notify(
   System.String notify,
   System.Object optional
);
```

IVsaItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "E0C0FFE5-7eea-4ee5-b7e4-
0080c7eb0b74") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface IVsaItem
```

Public Properties

```
boolean get_IsDirty();
Microsoft.Vsa.VsaItemType get_ItemType();
System.String get_Name();
void set_Name(
   System.String value
);
```

Public Methods

```
System.Object GetOption(
   System.String name
);
void SetOption(
   System.String name,
   System.Object value
);
```

IVsaItems Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "0AB1EB6A-12BD-44d0-B941-
0580ADFC73DE") */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public interface IVsaItems
   implements System.Collections.IEnumerable
```

Public Properties

```
int get_Count();
Microsoft.Vsa.IVsaItem get_Item(
   int index
);
Microsoft.Vsa.IVsaItem get_Item(
   System.String name
);
```

Public Methods

```
Microsoft.Vsa.IVsaItem CreateItem(
   System.String name,
   Microsoft.Vsa.VsaItemType itemType,
   Microsoft.Vsa.VsaItemFlag itemFlag
);
void Remove(
   int index
);
void Remove(
   System.String name
);
```

IVsaPersistSite Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "E0C0FFE3-7eea-4ee2-b7e4-
0080c7eb0b74") */
public interface IVsaPersistSite
```

Public Methods

```
System.String LoadElement(
   System.String name
);
void SaveElement(
   System.String name,
   System.String source
);
```

IVsaReferenceItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "E0C0FFE6-7eea-4ee5-b7e4-
0080c7eb0b74") */
public interface IVsaReferenceItem
   implements Microsoft.Vsa.IVsaItem
```

Public Properties

```
System.String get_AssemblyName();
void set_AssemblyName(
   System.String value
);
```

IVsaSite Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "E0C0FFE2-7eea-4ee2-b7e4-
0080c7eb0b74") */
public interface IVsaSite
```

Public Methods

```
void GetCompiledState(
   System.Byte[]& pe,
   System.Byte[]& debugInfo
);
System.Object GetEventSourceInstance(
   System.String itemName,
   System.String eventSourceName
);
System.Object GetGlobalInstance(
   System.String name
);
void Notify(
   System.String notify,
   System.Object info
);
boolean OnCompilerError(
   Microsoft.Vsa.IVsaError error
);
```

VsaError Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum VsaError
```

Members

AppDomainCannotBeSet

AppDomainInvalid

ApplicationBaseCannotBeSet

ApplicationBaseInvalid

AssemblyExpected

AssemblyNameInvalid

BadAssembly

BrowserNotExist

CachedAssemblyInvalid

CallbackUnexpected

CannotAttachToWebServer

CodeDOMNotAvailable

CompiledStateNotFound

DebuggeeNotStarted

DebugInfoNotSupported

ElementNameInvalid

ElementNotFound

EngineBusy

EngineCannotClose

EngineCannotReset

EngineClosed

EngineEmpty

EngineInitialized

EngineNameInUse

EngineNameInvalid

EngineNameNotSet

EngineNotCompiled

EngineNotExist

EngineNotInitialized

EngineNotRunning

EngineRunning

EventSourceInvalid

EventSourceNameInUse

EventSourceNameInvalid

EventSourceNotFound

EventSourceTypeInvalid

FileFormatUnsupported

FileTypeUnknown

GetCompiledStateFailed

GlobalInstanceInvalid

GlobalInstanceTypeInvalid

InternalCompilerError

ItemCannotBeRemoved

ItemCannotBeRenamed

ItemFlagNotSupported

ItemNameInUse

ItemNameInvalid

ItemNotFound

ItemTypeNotSupported

LCIDNotSupported

LoadElementFailed

MissingPdb

MissingSource

NameTooLong

NotClientSideAndNoUrl

NotificationInvalid

NotInitCompleted

OptionInvalid

OptionNotSupported

ProcNameInUse

ProcNameInvalid

RevokeFailed

RootMonikerAlreadySet

RootMonikerInUse

RootMonikerInvalid

RootMonikerNotSet

RootMonikerProtocolInvalid

RootNamespaceInvalid

RootNamespaceNotSet

SaveCompiledStateFailed

SaveElementFailed

SiteAlreadySet

SiteInvalid

SiteNotSet

SourceItemNotAvailable

SourceMonikerNotAvailable

UnknownError

URLInvalid

VsaServerDown

VsaException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class VsaException
   extends System.Runtime.InteropServices.ExternalException
```

Public Constructors

```
public VsaException(
   Microsoft.Vsa.VsaError error
);
public VsaException(
   Microsoft.Vsa.VsaError error,
   System.String message
);
public VsaException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public VsaException(
   Microsoft.Vsa.VsaError error,
   System.String message,
   System.Exception innerException
);
```

Public Properties

```
public Microsoft.Vsa.VsaError get_ErrorCode();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public System.String ToString();
```

VsaIDEMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum VsaIDEMode
```

Members

Break

Design

Run

VsaItemFlag Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum VsaItemFlag
```

Members

Class

Module

None

VsaItemType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum VsaItemType
```

Members

AppGlobal

Code

Reference

VsaLoader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class VsaLoader
   extends System.Object
   implements Microsoft.Vsa.IVsaEngine
```

Public Constructors

```
public VsaLoader();
```

Public Properties

```
public final System.Reflection.Assembly get_Assembly();
public final System.Security.Policy.Evidence get_Evidence();
public final void set_Evidence(
   System.Security.Policy.Evidence value
);
public final boolean get_GenerateDebugInfo();
public final void set_GenerateDebugInfo(
   boolean value
);
public final boolean get_IsCompiled();
public final boolean get_IsDirty();
public final boolean get_IsRunning();
public final Microsoft.Vsa.IVsaItems get_Items();
public final System.String get_Language();
public final int get_LCID();
public final void set_LCID(
   int value
);
public final System.String get_Name();
public final void set_Name(
   System.String value
);
public final System.String get_RootMoniker();
public final void set_RootMoniker(
   System.String value
);
public final System.String get_RootNamespace();
public final void set_RootNamespace(
   System.String value
);
public final Microsoft.Vsa.IVsaSite get_Site();
public final void set_Site(
   Microsoft.Vsa.IVsaSite value
);
public final System.String get_Version();
```

Public Methods

```
public final void Close();
public final boolean Compile();
public final System.Object GetOption(
   System.String name
);
```

```
public final void InitNew();
public final boolean IsValidIdentifier(
   System.String identifier
);
public final void LoadSourceState(
   Microsoft.Vsa.IVsaPersistSite Site
);
public final void Reset();
public final void RevokeCache();
public final void Run();
public final void SaveCompiledState(
   System.Byte[]& PE,
   System.Byte[]& PDB
);
public final void SaveSourceState(
   Microsoft.Vsa.IVsaPersistSite Site
);
public final void SetOption(
   System.String name,
   System.Object value
);
```

VsaModule Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class VsaModule
   extends System.Attribute
```

Public Constructors

```
public VsaModule(
   boolean bIsVsaModule
);
```

Public Properties

```
public boolean get_IsVsaModule();
public void set_IsVsaModule(
   boolean value
);
```

Microsoft.Win32 Namespace (Visual J# Syntax)

Classes

PowerModeChangedEventArgs

Registry

RegistryKey

SessionEndedEventArgs

SessionEndingEventArgs

SystemEvents

TimerElapsedEventArgs

UserPreferenceChangedEventArgs

UserPreferenceChangingEventArgs

Delegates

PowerModeChangedEventHandler

SessionEndedEventHandler

SessionEndingEventHandler

TimerElapsedEventHandler

UserPreferenceChangedEventHandler

UserPreferenceChangingEventHandler

Enumerations

PowerModes

RegistryHive

SessionEndReasons

UserPreferenceCategory

PowerModeChangedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PowerModeChangedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public PowerModeChangedEventArgs(
   Microsoft.Win32.PowerModes mode
);
```

Public Properties

```
public Microsoft.Win32.PowerModes get_Mode();
```

PowerModeChangedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void PowerModeChangedEventHandler(
   System.Object sender,
   Microsoft.Win32.PowerModeChangedEventArgs e
);
```

PowerModes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PowerModes
```

Members

Resume

StatusChange

Suspend

Registry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Registry
   extends System.Object
```

Public Fields

```
public static final Microsoft.Win32.RegistryKey ClassesRoot;
public static final Microsoft.Win32.RegistryKey CurrentConfig;
public static final Microsoft.Win32.RegistryKey CurrentUser;
public static final Microsoft.Win32.RegistryKey DynData;
public static final Microsoft.Win32.RegistryKey LocalMachine;
public static final Microsoft.Win32.RegistryKey PerformanceData;
public static final Microsoft.Win32.RegistryKey Users;
```

RegistryHive Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum RegistryHive
```

Members

ClassesRoot

CurrentConfig

CurrentUser

DynData

LocalMachine

PerformanceData

Users

RegistryKey Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RegistryKey
   extends System.MarshalByRefObject
   implements System.IDisposable
```

Public Properties

```
public System.String get_Name();
public int get_SubKeyCount();
public int get_ValueCount();
```

Public Methods

```
public void Close();
public Microsoft.Win32.RegistryKey CreateSubKey(
   System.String subkey
);
public void DeleteSubKey(
   System.String subkey
);
public void DeleteSubKey(
   System.String subkey,
   boolean throwOnMissingSubKey
);
public void DeleteSubKeyTree(
   System.String subkey
);
public void DeleteValue(
   System.String name
);
public void DeleteValue(
   System.String name,
   boolean throwOnMissingValue
);
public void Flush();
public System.String[] GetSubKeyNames();
public System.Object GetValue(
   System.String name
);
public System.Object GetValue(
   System.String name,
   System.Object defaultValue
);
public System.String[] GetValueNames();
public static Microsoft.Win32.RegistryKey OpenRemoteBaseKey(
   Microsoft.Win32.RegistryHive hKey,
   System.String machineName
);
public Microsoft.Win32.RegistryKey OpenSubKey(
   System.String name
);
public Microsoft.Win32.RegistryKey OpenSubKey(
   System.String name,
   boolean writable
);
public void SetValue(
```

```
    System.String name,
    System.Object value
);
public System.String ToString();
```

Protected Methods

```
protected void Finalize();
```

SessionEndedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SessionEndedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public SessionEndedEventArgs(
   Microsoft.Win32.SessionEndReasons reason
);
```

Public Properties

```
public Microsoft.Win32.SessionEndReasons get_Reason();
```

SessionEndedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void SessionEndedEventHandler(
   System.Object sender,
   Microsoft.Win32.SessionEndedEventArgs e
);
```

SessionEndingEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SessionEndingEventArgs
   extends System.EventArgs
```

Public Constructors

```
public SessionEndingEventArgs(
   Microsoft.Win32.SessionEndReasons reason
);
```

Public Properties

```
public boolean get_Cancel();
public void set_Cancel(
   boolean value
);
public Microsoft.Win32.SessionEndReasons get_Reason();
```

SessionEndingEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void SessionEndingEventHandler(
   System.Object sender,
   Microsoft.Win32.SessionEndingEventArgs e
);
```

SessionEndReasons Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SessionEndReasons
```

Members

Logoff

SystemShutdown

SystemEvents Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SystemEvents
   extends System.Object
```

Public Methods

```
public static System.IntPtr CreateTimer(
   int interval
);
public static void InvokeOnEventsThread(
   System.Delegate method
);
public static void KillTimer(
   System.IntPtr timerId
);
```

Public Events

```
public static void add_DisplaySettingsChanged(
   System.EventHandler value
);
public static void remove_DisplaySettingsChanged(
   System.EventHandler value
);
public static void add_EventsThreadShutdown(
   System.EventHandler value
);
public static void remove_EventsThreadShutdown(
   System.EventHandler value
);
public static void add_InstalledFontsChanged(
   System.EventHandler value
);
public static void remove_InstalledFontsChanged(
   System.EventHandler value
);
public static void add_LowMemory(
   System.EventHandler value
);
public static void remove_LowMemory(
   System.EventHandler value
);
public static void add_PaletteChanged(
   System.EventHandler value
);
public static void remove_PaletteChanged(
   System.EventHandler value
);
public static void add_PowerModeChanged(
   Microsoft.Win32.PowerModeChangedEventHandler value
);
public static void remove_PowerModeChanged(
   Microsoft.Win32.PowerModeChangedEventHandler value
);
public static void add_SessionEnded(
   Microsoft.Win32.SessionEndedEventHandler value
```

```
);
public static void remove_SessionEnded(
   Microsoft.Win32.SessionEndedEventHandler value
);
public static void add_SessionEnding(
   Microsoft.Win32.SessionEndingEventHandler value
);
public static void remove_SessionEnding(
   Microsoft.Win32.SessionEndingEventHandler value
);
public static void add_TimeChanged(
   System.EventHandler value
);
public static void remove_TimeChanged(
   System.EventHandler value
);
public static void add_TimerElapsed(
   Microsoft.Win32.TimerElapsedEventHandler value
);
public static void remove_TimerElapsed(
   Microsoft.Win32.TimerElapsedEventHandler value
);
public static void add_UserPreferenceChanged(
   Microsoft.Win32.UserPreferenceChangedEventHandler value
);
public static void remove_UserPreferenceChanged(
   Microsoft.Win32.UserPreferenceChangedEventHandler value
);
public static void add_UserPreferenceChanging(
   Microsoft.Win32.UserPreferenceChangingEventHandler value
);
public static void remove_UserPreferenceChanging(
   Microsoft.Win32.UserPreferenceChangingEventHandler value
);
```

TimerElapsedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class TimerElapsedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public TimerElapsedEventArgs(
   System.IntPtr timerId
);
```

Public Properties

```
public System.IntPtr get_TimerId();
```

TimerElapsedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void TimerElapsedEventHandler(
   System.Object sender,
   Microsoft.Win32.TimerElapsedEventArgs e
);
```

UserPreferenceCategory Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum UserPreferenceCategory
```

Members

Accessibility

Color

Desktop

General

Icon

Keyboard

Locale

Menu

Mouse

Policy

Power

Screensaver

Window

UserPreferenceChangedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UserPreferenceChangedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public UserPreferenceChangedEventArgs(
   Microsoft.Win32.UserPreferenceCategory category
);
```

Public Properties

```
public Microsoft.Win32.UserPreferenceCategory get_Category();
```

UserPreferenceChangedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void UserPreferenceChangedEventHandler(
   System.Object sender,
   Microsoft.Win32.UserPreferenceChangedEventArgs e
);
```

UserPreferenceChangingEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UserPreferenceChangingEventArgs
   extends System.EventArgs
```

Public Constructors

```
public UserPreferenceChangingEventArgs(
   Microsoft.Win32.UserPreferenceCategory category
);
```

Public Properties

```
public Microsoft.Win32.UserPreferenceCategory get_Category();
```

UserPreferenceChangingEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void UserPreferenceChangingEventHandler(
   System.Object sender,
   Microsoft.Win32.UserPreferenceChangingEventArgs e
);
```

System Namespace (Visual J# Syntax)

Classes

Activator

AppDomain

AppDomainSetup

AppDomainUnloadedException

ApplicationException

ArgumentException

ArgumentNullException

ArgumentOutOfRangeException

ArithmeticException

Array

ArrayTypeMismatchException

AssemblyLoadEventArgs

Attribute

AttributeUsageAttribute

BadImageFormatException

BitConverter

Buffer

CannotUnloadAppDomainException

CharEnumerator

CLSCompliantAttribute

Console

ContextBoundObject

ContextMarshalException

ContextStaticAttribute

Convert

DBNull

Delegate

DivideByZeroException

DllNotFoundException

DuplicateWaitObjectException

EntryPointNotFoundException

Environment

EventArgs

Exception

ExecutionEngineException

FieldAccessException

FlagsAttribute

FormatException

GC

IndexOutOfRangeException

InvalidCastException

InvalidOperationException

InvalidProgramException

LoaderOptimizationAttribute

LocalDataStoreSlot

MarshalByRefObject

Math

MemberAccessException

MethodAccessException

MissingFieldException

MissingMemberException

MissingMethodException

MTAThreadAttribute

MulticastDelegate

MulticastNotSupportedException

NonSerializedAttribute

NotFiniteNumberException

NotImplementedException

NotSupportedException

NullReferenceException

Object

ObjectDisposedException

ObsoleteAttribute

OperatingSystem

OutOfMemoryException

OverflowException

ParamArrayAttribute

PlatformNotSupportedException

Random

RankException

ResolveEventArgs

SerializableAttribute

StackOverflowException

STAThreadAttribute

String

SystemException

ThreadStaticAttribute

TimeZone

Type

TypeInitializationException

TypeLoadException

TypeUnloadedException

UnauthorizedAccessException

UnhandledExceptionEventArgs

Uri

UriBuilder

UriFormatException

ValueType

Version

WeakReference

Interfaces

_AppDomain

IAppDomainSetup

IAsyncResult

ICloneable

IComparable

IConvertible

ICustomFormatter

IDisposable

IFormatProvider

IFormattable

IServiceProvider

Structures

ArgIterator

Boolean

Byte

Char

DateTime

Decimal

Double

Guid

Int16

Int32

Int64

IntPtr

RuntimeArgumentHandle

RuntimeFieldHandle

RuntimeMethodHandle

RuntimeTypeHandle

SByte

Single

TimeSpan

TypedReference

UInt16

UInt32

UInt64

UIntPtr

Void

Delegates

AssemblyLoadEventHandler

AsyncCallback

CrossAppDomainDelegate

EventHandler

ResolveEventHandler

UnhandledExceptionEventHandler

Enumerations

AttributeTargets

DayOfWeek

Environment.SpecialFolder

LoaderOptimization

PlatformID

TypeCode

UriHostNameType

UriPartial

_AppDomain Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "05F696DC-2B29-3663-AD8B-
C4389CF2A713") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsDual)
*/
public interface _AppDomain
```

Public Properties

```
System.String get_BaseDirectory();
System.String get_DynamicDirectory();
System.Security.Policy.Evidence get_Evidence();
System.String get_FriendlyName();
System.String get_RelativeSearchPath();
boolean get_ShadowCopyFiles();
```

Public Methods

```
void AppendPrivatePath(
   System.String path
);
void ClearPrivatePath();
void ClearShadowCopyPath();
System.Runtime.Remoting.ObjectHandle CreateInstance(
   System.String assemblyName,
   System.String typeName
);
System.Runtime.Remoting.ObjectHandle CreateInstance(
   System.String assemblyName,
   System.String typeName,
   System.Object[] activationAttributes
);
System.Runtime.Remoting.ObjectHandle CreateInstance(
   System.String assemblyName,
   System.String typeName,
   boolean ignoreCase,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Object[] args,
   System.Globalization.CultureInfo culture,
   System.Object[] activationAttributes,
   System.Security.Policy.Evidence securityAttributes
);
System.Runtime.Remoting.ObjectHandle CreateInstanceFrom(
   System.String assemblyFile,
   System.String typeName
);
System.Runtime.Remoting.ObjectHandle CreateInstanceFrom(
   System.String assemblyFile,
   System.String typeName,
   System.Object[] activationAttributes
);
System.Runtime.Remoting.ObjectHandle CreateInstanceFrom(
   System.String assemblyFile,
   System.String typeName,
```

```
    boolean ignoreCase,
    System.Reflection.BindingFlags bindingAttr,
    System.Reflection.Binder binder,
    System.Object[] args,
    System.Globalization.CultureInfo culture,
    System.Object[] activationAttributes,
    System.Security.Policy.Evidence securityAttributes
);
System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
    System.Reflection.AssemblyName name,
    System.Reflection.Emit.AssemblyBuilderAccess access
);
System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
    System.Reflection.AssemblyName name,
    System.Reflection.Emit.AssemblyBuilderAccess access,
    System.Security.Policy.Evidence evidence
);
System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
    System.Reflection.AssemblyName name,
    System.Reflection.Emit.AssemblyBuilderAccess access,
    System.String dir
);
System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
    System.Reflection.AssemblyName name,
    System.Reflection.Emit.AssemblyBuilderAccess access,
    System.String dir,
    System.Security.Policy.Evidence evidence
);
System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
    System.Reflection.AssemblyName name,
    System.Reflection.Emit.AssemblyBuilderAccess access,
    System.Security.PermissionSet requiredPermissions,
    System.Security.PermissionSet optionalPermissions,
    System.Security.PermissionSet refusedPermissions
);
System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
    System.Reflection.AssemblyName name,
    System.Reflection.Emit.AssemblyBuilderAccess access,
    System.Security.Policy.Evidence evidence,
    System.Security.PermissionSet requiredPermissions,
    System.Security.PermissionSet optionalPermissions,
    System.Security.PermissionSet refusedPermissions
);
System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
    System.Reflection.AssemblyName name,
    System.Reflection.Emit.AssemblyBuilderAccess access,
    System.String dir,
    System.Security.PermissionSet requiredPermissions,
    System.Security.PermissionSet optionalPermissions,
    System.Security.PermissionSet refusedPermissions
);
System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
    System.Reflection.AssemblyName name,
    System.Reflection.Emit.AssemblyBuilderAccess access,
    System.String dir,
    System.Security.Policy.Evidence evidence,
    System.Security.PermissionSet requiredPermissions,
    System.Security.PermissionSet optionalPermissions,
    System.Security.PermissionSet refusedPermissions
);
System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
    System.Reflection.AssemblyName name,
    System.Reflection.Emit.AssemblyBuilderAccess access,
    System.String dir,
    System.Security.Policy.Evidence evidence,
    System.Security.PermissionSet requiredPermissions,
    System.Security.PermissionSet optionalPermissions,
    System.Security.PermissionSet refusedPermissions,
```

```
   boolean isSynchronized
);
void DoCallBack(
   System.CrossAppDomainDelegate theDelegate
);
boolean Equals(
   System.Object other
);
int ExecuteAssembly(
   System.String assemblyFile
);
int ExecuteAssembly(
   System.String assemblyFile,
   System.Security.Policy.Evidence assemblySecurity
);
int ExecuteAssembly(
   System.String assemblyFile,
   System.Security.Policy.Evidence assemblySecurity,
   System.String[] args
);
System.Reflection.Assembly[] GetAssemblies();
System.Object GetData(
   System.String name
);
int GetHashCode();
System.Object GetLifetimeService();
System.Type GetType();
System.Object InitializeLifetimeService();
System.Reflection.Assembly Load(
   ubyte[] rawAssembly
);
System.Reflection.Assembly Load(
   System.Reflection.AssemblyName assemblyRef
);
System.Reflection.Assembly Load(
   System.String assemblyString
);
System.Reflection.Assembly Load(
   ubyte[] rawAssembly,
   ubyte[] rawSymbolStore
);
System.Reflection.Assembly Load(
   System.Reflection.AssemblyName assemblyRef,
   System.Security.Policy.Evidence assemblySecurity
);
System.Reflection.Assembly Load(
   System.String assemblyString,
   System.Security.Policy.Evidence assemblySecurity
);
System.Reflection.Assembly Load(
   ubyte[] rawAssembly,
   ubyte[] rawSymbolStore,
   System.Security.Policy.Evidence securityEvidence
);
void SetAppDomainPolicy(
   System.Security.Policy.PolicyLevel domainPolicy
);
void SetCachePath(
   System.String s
);
void SetData(
   System.String name,
   System.Object data
);
void SetPrincipalPolicy(
   System.Security.Principal.PrincipalPolicy policy
);
void SetShadowCopyPath(
```

```
   System.String s
);
void SetThreadPrincipal(
   System.Security.Principal.IPrincipal principal
);
System.String ToString();
```

Public Events

```
void add_AssemblyLoad(
   System.AssemblyLoadEventHandler value
);
void remove_AssemblyLoad(
   System.AssemblyLoadEventHandler value
);
void add_AssemblyResolve(
   System.ResolveEventHandler value
);
void remove_AssemblyResolve(
   System.ResolveEventHandler value
);
void add_DomainUnload(
   System.EventHandler value
);
void remove_DomainUnload(
   System.EventHandler value
);
void add_ProcessExit(
   System.EventHandler value
);
void remove_ProcessExit(
   System.EventHandler value
);
void add_ResourceResolve(
   System.ResolveEventHandler value
);
void remove_ResourceResolve(
   System.ResolveEventHandler value
);
void add_TypeResolve(
   System.ResolveEventHandler value
);
void remove_TypeResolve(
   System.ResolveEventHandler value
);
void add_UnhandledException(
   System.UnhandledExceptionEventHandler value
);
void remove_UnhandledException(
   System.UnhandledExceptionEventHandler value
);
```

Activator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Activator
   extends System.Object
```

Public Methods

```
public static System.Runtime.Remoting.ObjectHandle CreateComInstanceFrom(
   System.String assemblyName,
   System.String typeName
);
public static System.Object CreateInstance(
   System.Type type
);
public static System.Object CreateInstance(
   System.Type type,
   boolean nonPublic
);
public static System.Object CreateInstance(
   System.Type type,
   System.Object[] args
);
public static System.Runtime.Remoting.ObjectHandle CreateInstance(
   System.String assemblyName,
   System.String typeName
);
public static System.Object CreateInstance(
   System.Type type,
   System.Object[] args,
   System.Object[] activationAttributes
);
public static System.Runtime.Remoting.ObjectHandle CreateInstance(
   System.String assemblyName,
   System.String typeName,
   System.Object[] activationAttributes
);
public static System.Object CreateInstance(
   System.Type type,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Object[] args,
   System.Globalization.CultureInfo culture
);
public static System.Object CreateInstance(
   System.Type type,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Object[] args,
   System.Globalization.CultureInfo culture,
   System.Object[] activationAttributes
);
public static System.Runtime.Remoting.ObjectHandle CreateInstance(
   System.String assemblyName,
   System.String typeName,
   boolean ignoreCase,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Object[] args,
   System.Globalization.CultureInfo culture,
```

```
   System.Object[] activationAttributes,
   System.Security.Policy.Evidence securityInfo
);
public static System.Runtime.Remoting.ObjectHandle CreateInstanceFrom(
   System.String assemblyFile,
   System.String typeName
);
public static System.Runtime.Remoting.ObjectHandle CreateInstanceFrom(
   System.String assemblyFile,
   System.String typeName,
   System.Object[] activationAttributes
);
public static System.Runtime.Remoting.ObjectHandle CreateInstanceFrom(
   System.String assemblyFile,
   System.String typeName,
   boolean ignoreCase,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Object[] args,
   System.Globalization.CultureInfo culture,
   System.Object[] activationAttributes,
   System.Security.Policy.Evidence securityInfo
);
public static System.Object GetObject(
   System.Type type,
   System.String url
);
public static System.Object GetObject(
   System.Type type,
   System.String url,
   System.Object state
);
```

AppDomain Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = None) */
public class AppDomain
   extends System.MarshalByRefObject
   implements System._AppDomain, System.Security.IEvidenceFactory
```

Public Properties

```
public final System.String get_BaseDirectory();
public static System.AppDomain get_CurrentDomain();
public final System.String get_DynamicDirectory();
public final System.Security.Policy.Evidence get_Evidence();
public final System.String get_FriendlyName();
public final System.String get_RelativeSearchPath();
public System.AppDomainSetup get_SetupInformation();
public final boolean get_ShadowCopyFiles();
```

Public Methods

```
public final void AppendPrivatePath(
   System.String path
);
public final void ClearPrivatePath();
public final void ClearShadowCopyPath();
public System.Runtime.Remoting.ObjectHandle CreateComInstanceFrom(
   System.String assemblyName,
   System.String typeName
);
public static System.AppDomain CreateDomain(
   System.String friendlyName
);
public static System.AppDomain CreateDomain(
   System.String friendlyName,
   System.Security.Policy.Evidence securityInfo
);
public static System.AppDomain CreateDomain(
   System.String friendlyName,
   System.Security.Policy.Evidence securityInfo,
   System.AppDomainSetup info
);
public static System.AppDomain CreateDomain(
   System.String friendlyName,
   System.Security.Policy.Evidence securityInfo,
   System.String appBasePath,
   System.String appRelativeSearchPath,
   boolean shadowCopyFiles
);
public final System.Runtime.Remoting.ObjectHandle CreateInstance(
   System.String assemblyName,
   System.String typeName
);
public final System.Runtime.Remoting.ObjectHandle CreateInstance(
   System.String assemblyName,
   System.String typeName,
   System.Object[] activationAttributes
);
```

```
public final System.Runtime.Remoting.ObjectHandle CreateInstance(
   System.String assemblyName,
   System.String typeName,
   boolean ignoreCase,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Object[] args,
   System.Globalization.CultureInfo culture,
   System.Object[] activationAttributes,
   System.Security.Policy.Evidence securityAttributes
);
public System.Object CreateInstanceAndUnwrap(
   System.String assemblyName,
   System.String typeName
);
public System.Object CreateInstanceAndUnwrap(
   System.String assemblyName,
   System.String typeName,
   System.Object[] activationAttributes
);
public System.Object CreateInstanceAndUnwrap(
   System.String assemblyName,
   System.String typeName,
   boolean ignoreCase,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Object[] args,
   System.Globalization.CultureInfo culture,
   System.Object[] activationAttributes,
   System.Security.Policy.Evidence securityAttributes
);
public final System.Runtime.Remoting.ObjectHandle CreateInstanceFrom(
   System.String assemblyFile,
   System.String typeName
);
public final System.Runtime.Remoting.ObjectHandle CreateInstanceFrom(
   System.String assemblyFile,
   System.String typeName,
   System.Object[] activationAttributes
);
public final System.Runtime.Remoting.ObjectHandle CreateInstanceFrom(
   System.String assemblyFile,
   System.String typeName,
   boolean ignoreCase,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Object[] args,
   System.Globalization.CultureInfo culture,
   System.Object[] activationAttributes,
   System.Security.Policy.Evidence securityAttributes
);
public System.Object CreateInstanceFromAndUnwrap(
   System.String assemblyName,
   System.String typeName
);
public System.Object CreateInstanceFromAndUnwrap(
   System.String assemblyName,
   System.String typeName,
   System.Object[] activationAttributes
);
public System.Object CreateInstanceFromAndUnwrap(
   System.String assemblyName,
   System.String typeName,
   boolean ignoreCase,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Object[] args,
   System.Globalization.CultureInfo culture,
```

```
   System.Object[] activationAttributes,
   System.Security.Policy.Evidence securityAttributes
);
public final System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
   System.Reflection.AssemblyName name,
   System.Reflection.Emit.AssemblyBuilderAccess access
);
public final System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
   System.Reflection.AssemblyName name,
   System.Reflection.Emit.AssemblyBuilderAccess access,
   System.Security.Policy.Evidence evidence
);
public final System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
   System.Reflection.AssemblyName name,
   System.Reflection.Emit.AssemblyBuilderAccess access,
   System.String dir
);
public final System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
   System.Reflection.AssemblyName name,
   System.Reflection.Emit.AssemblyBuilderAccess access,
   System.String dir,
   System.Security.Policy.Evidence evidence
);
public final System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
   System.Reflection.AssemblyName name,
   System.Reflection.Emit.AssemblyBuilderAccess access,
   System.Security.PermissionSet requiredPermissions,
   System.Security.PermissionSet optionalPermissions,
   System.Security.PermissionSet refusedPermissions
);
public final System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
   System.Reflection.AssemblyName name,
   System.Reflection.Emit.AssemblyBuilderAccess access,
   System.Security.Policy.Evidence evidence,
   System.Security.PermissionSet requiredPermissions,
   System.Security.PermissionSet optionalPermissions,
   System.Security.PermissionSet refusedPermissions
);
public final System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
   System.Reflection.AssemblyName name,
   System.Reflection.Emit.AssemblyBuilderAccess access,
   System.String dir,
   System.Security.PermissionSet requiredPermissions,
   System.Security.PermissionSet optionalPermissions,
   System.Security.PermissionSet refusedPermissions
);
public final System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
   System.Reflection.AssemblyName name,
   System.Reflection.Emit.AssemblyBuilderAccess access,
   System.String dir,
   System.Security.Policy.Evidence evidence,
   System.Security.PermissionSet requiredPermissions,
   System.Security.PermissionSet optionalPermissions,
   System.Security.PermissionSet refusedPermissions
);
public final System.Reflection.Emit.AssemblyBuilder DefineDynamicAssembly(
   System.Reflection.AssemblyName name,
   System.Reflection.Emit.AssemblyBuilderAccess access,
   System.String dir,
   System.Security.Policy.Evidence evidence,
   System.Security.PermissionSet requiredPermissions,
   System.Security.PermissionSet optionalPermissions,
   System.Security.PermissionSet refusedPermissions,
   boolean isSynchronized
);
public final void DoCallBack(
   System.CrossAppDomainDelegate callBackDelegate
);
```

```
public final int ExecuteAssembly(
   System.String assemblyFile
);
public final int ExecuteAssembly(
   System.String assemblyFile,
   System.Security.Policy.Evidence assemblySecurity
);
public final int ExecuteAssembly(
   System.String assemblyFile,
   System.Security.Policy.Evidence assemblySecurity,
   System.String[] args
);
public final System.Reflection.Assembly[] GetAssemblies();
public static int GetCurrentThreadId();
public final System.Object GetData(
   System.String name
);
public final System.Type GetType();
public System.Object InitializeLifetimeService();
public boolean IsFinalizingForUnload();
public final System.Reflection.Assembly Load(
   ubyte[] rawAssembly
);
public final System.Reflection.Assembly Load(
   System.Reflection.AssemblyName assemblyRef
);
public final System.Reflection.Assembly Load(
   System.String assemblyString
);
public final System.Reflection.Assembly Load(
   ubyte[] rawAssembly,
   ubyte[] rawSymbolStore
);
public final System.Reflection.Assembly Load(
   System.Reflection.AssemblyName assemblyRef,
   System.Security.Policy.Evidence assemblySecurity
);
public final System.Reflection.Assembly Load(
   System.String assemblyString,
   System.Security.Policy.Evidence assemblySecurity
);
public final System.Reflection.Assembly Load(
   ubyte[] rawAssembly,
   ubyte[] rawSymbolStore,
   System.Security.Policy.Evidence securityEvidence
);
public final void SetAppDomainPolicy(
   System.Security.Policy.PolicyLevel domainPolicy
);
public final void SetCachePath(
   System.String path
);
public final void SetData(
   System.String name,
   System.Object data
);
public void SetDynamicBase(
   System.String path
);
public final void SetPrincipalPolicy(
   System.Security.Principal.PrincipalPolicy policy
);
public void SetShadowCopyFiles();
public final void SetShadowCopyPath(
   System.String path
);
public final void SetThreadPrincipal(
   System.Security.Principal.IPrincipal principal
```

```
);
public System.String ToString();
public static void Unload(
   System.AppDomain domain
);
```

Public Events

```
public final void add_AssemblyLoad(
   System.AssemblyLoadEventHandler value
);
public final void remove_AssemblyLoad(
   System.AssemblyLoadEventHandler value
);
public final void add_AssemblyResolve(
   System.ResolveEventHandler value
);
public final void remove_AssemblyResolve(
   System.ResolveEventHandler value
);
public final void add_DomainUnload(
   System.EventHandler value
);
public final void remove_DomainUnload(
   System.EventHandler value
);
public final void add_ProcessExit(
   System.EventHandler value
);
public final void remove_ProcessExit(
   System.EventHandler value
);
public final void add_ResourceResolve(
   System.ResolveEventHandler value
);
public final void remove_ResourceResolve(
   System.ResolveEventHandler value
);
public final void add_TypeResolve(
   System.ResolveEventHandler value
);
public final void remove_TypeResolve(
   System.ResolveEventHandler value
);
public final void add_UnhandledException(
   System.UnhandledExceptionEventHandler value
);
public final void remove_UnhandledException(
   System.UnhandledExceptionEventHandler value
);
```

AppDomainSetup Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = None) */
public class AppDomainSetup
   extends System.Object
   implements System.IAppDomainSetup
```

Public Constructors

```
public AppDomainSetup();
```

Public Properties

```
public final System.String get_ApplicationBase();
public final void set_ApplicationBase(
   System.String value
);
public final System.String get_ApplicationName();
public final void set_ApplicationName(
   System.String value
);
public final System.String get_CachePath();
public final void set_CachePath(
   System.String value
);
public final System.String get_ConfigurationFile();
public final void set_ConfigurationFile(
   System.String value
);
public boolean get_DisallowPublisherPolicy();
public void set_DisallowPublisherPolicy(
   boolean value
);
public final System.String get_DynamicBase();
public final void set_DynamicBase(
   System.String value
);
public final System.String get_LicenseFile();
public final void set_LicenseFile(
   System.String value
);
public System.LoaderOptimization get_LoaderOptimization();
public void set_LoaderOptimization(
   System.LoaderOptimization value
);
public final System.String get_PrivateBinPath();
public final void set_PrivateBinPath(
   System.String value
);
public final System.String get_PrivateBinPathProbe();
public final void set_PrivateBinPathProbe(
   System.String value
);
public final System.String get_ShadowCopyDirectories();
public final void set_ShadowCopyDirectories(
   System.String value
```

```
);
public final System.String get_ShadowCopyFiles();
public final void set_ShadowCopyFiles(
   System.String value
);
```

AppDomainUnloadedException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class AppDomainUnloadedException
   extends System.SystemException
```

Public Constructors

```
public AppDomainUnloadedException();
public AppDomainUnloadedException(
   System.String message
);
public AppDomainUnloadedException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected AppDomainUnloadedException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ApplicationException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ApplicationException
   extends System.Exception
```

Public Constructors

```
public ApplicationException();
public ApplicationException(
   System.String message
);
public ApplicationException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected ApplicationException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ArgIterator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ArgIterator
   extends System.ValueType
```

Public Constructors

```
public ArgIterator(
   System.RuntimeArgumentHandle arglist
);
```

Public Methods

```
public void End();
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public System.RuntimeTypeHandle GetNextArgType();
public int GetRemainingCount();
```

ArgumentException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ArgumentException
   extends System.SystemException
```

Public Constructors

```
public ArgumentException();
public ArgumentException(
   System.String message
);
public ArgumentException(
   System.String message,
   System.Exception innerException
);
public ArgumentException(
   System.String message,
   System.String paramName
);
public ArgumentException(
   System.String message,
   System.String paramName,
   System.Exception innerException
);
```

Public Properties

```
public System.String get_Message();
public System.String get_ParamName();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected ArgumentException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ArgumentNullException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ArgumentNullException
   extends System.ArgumentException
```

Public Constructors

```
public ArgumentNullException();
public ArgumentNullException(
   System.String paramName
);
public ArgumentNullException(
   System.String paramName,
   System.String message
);
```

Protected Constructors

```
protected ArgumentNullException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ArgumentOutOfRangeException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ArgumentOutOfRangeException
   extends System.ArgumentException
```

Public Constructors

```
public ArgumentOutOfRangeException();
public ArgumentOutOfRangeException(
   System.String paramName
);
public ArgumentOutOfRangeException(
   System.String paramName,
   System.String message
);
public ArgumentOutOfRangeException(
   System.String paramName,
   System.Object actualValue,
   System.String message
);
```

Public Properties

```
public System.Object get_ActualValue();
public System.String get_Message();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected ArgumentOutOfRangeException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ArithmeticException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ArithmeticException
   extends System.SystemException
```

Public Constructors

```
public ArithmeticException();
public ArithmeticException(
   System.String message
);
public ArithmeticException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected ArithmeticException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Array Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class Array
   extends System.Object
   implements System.ICloneable, System.Collections.IList, System.Collections.ICollection, Sy
stem.Collections.IEnumerable
```

Public Properties

```
public boolean get_IsFixedSize();
public boolean get_IsReadOnly();
public boolean get_IsSynchronized();
public int get_Length();
public int get_Rank();
public System.Object get_SyncRoot();
```

Public Methods

```
public static int BinarySearch(
   System.Array array,
   System.Object value
);
public static int BinarySearch(
   System.Array array,
   System.Object value,
   System.Collections.IComparer comparer
);
public static int BinarySearch(
   System.Array array,
   int index,
   int length,
   System.Object value
);
public static int BinarySearch(
   System.Array array,
   int index,
   int length,
   System.Object value,
   System.Collections.IComparer comparer
);
public static void Clear(
   System.Array array,
   int index,
   int length
);
public System.Object Clone();
public static void Copy(
   System.Array sourceArray,
   System.Array destinationArray,
   int length
);
public static void Copy(
   System.Array sourceArray,
   int sourceIndex,
   System.Array destinationArray,
   int destinationIndex,
```

```
   int length
);
public void CopyTo(
   System.Array array,
   int index
);
public static System.Array CreateInstance(
   System.Type elementType,
   int length
);
public static System.Array CreateInstance(
   System.Type elementType,
   int[] lengths
);
public static System.Array CreateInstance(
   System.Type elementType,
   int length1,
   int length2
);
public static System.Array CreateInstance(
   System.Type elementType,
   int[] lengths,
   int[] lowerBounds
);
public static System.Array CreateInstance(
   System.Type elementType,
   int length1,
   int length2,
   int length3
);
public System.Collections.IEnumerator GetEnumerator();
public int GetLength(
   int dimension
);
public int GetLowerBound(
   int dimension
);
public int GetUpperBound(
   int dimension
);
public System.Object GetValue(
   int index
);
public System.Object GetValue(
   int[] indices
);
public System.Object GetValue(
   int index1,
   int index2
);
public System.Object GetValue(
   int index1,
   int index2,
   int index3
);
public static int IndexOf(
   System.Array array,
   System.Object value
);
public static int IndexOf(
   System.Array array,
   System.Object value,
   int startIndex
);
public static int IndexOf(
   System.Array array,
   System.Object value,
   int startIndex,
```

```
      int count
   );
   public void Initialize();
   public static int LastIndexOf(
      System.Array array,
      System.Object value
   );
   public static int LastIndexOf(
      System.Array array,
      System.Object value,
      int startIndex
   );
   public static int LastIndexOf(
      System.Array array,
      System.Object value,
      int startIndex,
      int count
   );
   public static void Reverse(
      System.Array array
   );
   public static void Reverse(
      System.Array array,
      int index,
      int length
   );
   public void SetValue(
      System.Object value,
      int index
   );
   public void SetValue(
      System.Object value,
      int[] indices
   );
   public void SetValue(
      System.Object value,
      int index1,
      int index2
   );
   public void SetValue(
      System.Object value,
      int index1,
      int index2,
      int index3
   );
   public static void Sort(
      System.Array array
   );
   public static void Sort(
      System.Array keys,
      System.Array items
   );
   public static void Sort(
      System.Array array,
      System.Collections.IComparer comparer
   );
   public static void Sort(
      System.Array array,
      int index,
      int length
   );
   public static void Sort(
      System.Array keys,
      System.Array items,
      System.Collections.IComparer comparer
   );
   public static void Sort(
      System.Array array,
```

```
    int index,
    int length,
    System.Collections.IComparer comparer
);
public static void Sort(
    System.Array keys,
    System.Array items,
    int index,
    int length
);
public static void Sort(
    System.Array keys,
    System.Array items,
    int index,
    int length,
    System.Collections.IComparer comparer
);
```

ArrayTypeMismatchException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ArrayTypeMismatchException
   extends System.SystemException
```

Public Constructors

```
public ArrayTypeMismatchException();
public ArrayTypeMismatchException(
   System.String message
);
public ArrayTypeMismatchException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected ArrayTypeMismatchException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

AssemblyLoadEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AssemblyLoadEventArgs
   extends System.EventArgs
```

Public Constructors

```
public AssemblyLoadEventArgs(
   System.Reflection.Assembly loadedAssembly
);
```

Public Properties

```
public System.Reflection.Assembly get_LoadedAssembly();
```

AssemblyLoadEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void AssemblyLoadEventHandler(
   System.Object sender,
   System.AssemblyLoadEventArgs args
);
```

AsyncCallback Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void AsyncCallback(
   System.IAsyncResult ar
);
```

Attribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public abstract class Attribute
   extends System.Object
```

Public Properties

```
public System.Object get_TypeId();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public static System.Attribute GetCustomAttribute(
   System.Reflection.Assembly element,
   System.Type attributeType
);
public static System.Attribute GetCustomAttribute(
   System.Reflection.MemberInfo element,
   System.Type attributeType
);
public static System.Attribute GetCustomAttribute(
   System.Reflection.Module element,
   System.Type attributeType
);
public static System.Attribute GetCustomAttribute(
   System.Reflection.ParameterInfo element,
   System.Type attributeType
);
public static System.Attribute GetCustomAttribute(
   System.Reflection.Assembly element,
   System.Type attributeType,
   boolean inherit
);
public static System.Attribute GetCustomAttribute(
   System.Reflection.MemberInfo element,
   System.Type attributeType,
   boolean inherit
);
public static System.Attribute GetCustomAttribute(
   System.Reflection.Module element,
   System.Type attributeType,
   boolean inherit
);
public static System.Attribute GetCustomAttribute(
   System.Reflection.ParameterInfo element,
   System.Type attributeType,
   boolean inherit
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.Assembly element
);
public static System.Attribute[] GetCustomAttributes(
```

```
   System.Reflection.MemberInfo element
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.Module element
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.ParameterInfo element
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.Assembly element,
   boolean inherit
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.Assembly element,
   System.Type attributeType
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.MemberInfo element,
   boolean inherit
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.MemberInfo element,
   System.Type type
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.Module element,
   boolean inherit
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.Module element,
   System.Type attributeType
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.ParameterInfo element,
   boolean inherit
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.ParameterInfo element,
   System.Type attributeType
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.Assembly element,
   System.Type attributeType,
   boolean inherit
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.MemberInfo element,
   System.Type type,
   boolean inherit
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.Module element,
   System.Type attributeType,
   boolean inherit
);
public static System.Attribute[] GetCustomAttributes(
   System.Reflection.ParameterInfo element,
   System.Type attributeType,
   boolean inherit
);
public int GetHashCode();
public boolean IsDefaultAttribute();
public static boolean IsDefined(
   System.Reflection.Assembly element,
   System.Type attributeType
);
public static boolean IsDefined(
   System.Reflection.MemberInfo element,
```

```
      System.Type attributeType
   );
   public static boolean IsDefined(
      System.Reflection.Module element,
      System.Type attributeType
   );
   public static boolean IsDefined(
      System.Reflection.ParameterInfo element,
      System.Type attributeType
   );
   public static boolean IsDefined(
      System.Reflection.Assembly element,
      System.Type attributeType,
      boolean inherit
   );
   public static boolean IsDefined(
      System.Reflection.MemberInfo element,
      System.Type attributeType,
      boolean inherit
   );
   public static boolean IsDefined(
      System.Reflection.Module element,
      System.Type attributeType,
      boolean inherit
   );
   public static boolean IsDefined(
      System.Reflection.ParameterInfo element,
      System.Type attributeType,
      boolean inherit
   );
   public boolean Match(
      System.Object obj
   );
```

Protected Constructors

```
   protected Attribute();
```

AttributeTargets Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum AttributeTargets
```

Members

All

Assembly

Class

Constructor

Delegate

Enum

Event

Field

Interface

Method

Module

Parameter

Property

ReturnValue

Struct

AttributeUsageAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class AttributeUsageAttribute
   extends System.Attribute
```

Public Constructors

```
public AttributeUsageAttribute(
   System.AttributeTargets validOn
);
```

Public Properties

```
public boolean get_AllowMultiple();
public void set_AllowMultiple(
   boolean value
);
public boolean get_Inherited();
public void set_Inherited(
   boolean value
);
public System.AttributeTargets get_ValidOn();
```

BadImageFormatException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class BadImageFormatException
   extends System.SystemException
```

Public Constructors

```
public BadImageFormatException();
public BadImageFormatException(
   System.String message
);
public BadImageFormatException(
   System.String message,
   System.Exception inner
);
public BadImageFormatException(
   System.String message,
   System.String fileName
);
public BadImageFormatException(
   System.String message,
   System.String fileName,
   System.Exception inner
);
```

Public Properties

```
public System.String get_FileName();
public System.String get_FusionLog();
public System.String get_Message();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public System.String ToString();
```

Protected Constructors

```
protected BadImageFormatException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

BitConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BitConverter
   extends System.Object
```

Public Methods

```
public static long DoubleToInt64Bits(
   double value
);
public static ubyte[] GetBytes(
   boolean value
);
public static ubyte[] GetBytes(
   char value
);
public static ubyte[] GetBytes(
   double value
);
public static ubyte[] GetBytes(
   short value
);
public static ubyte[] GetBytes(
   int value
);
public static ubyte[] GetBytes(
   long value
);
public static ubyte[] GetBytes(
   float value
);
public static double Int64BitsToDouble(
   long value
);
public static boolean ToBoolean(
   ubyte[] value,
   int startIndex
);
public static char ToChar(
   ubyte[] value,
   int startIndex
);
public static double ToDouble(
   ubyte[] value,
   int startIndex
);
public static short ToInt16(
   ubyte[] value,
   int startIndex
);
public static int ToInt32(
   ubyte[] value,
   int startIndex
);
public static long ToInt64(
   ubyte[] value,
   int startIndex
);
public static float ToSingle(
```

```
      ubyte[] value,
      int startIndex
   );
   public static System.String ToString(
      ubyte[] value
   );
   public static System.String ToString(
      ubyte[] value,
      int startIndex
   );
   public static System.String ToString(
      ubyte[] value,
      int startIndex,
      int length
   );
```

Public Fields

```
   public static final boolean IsLittleEndian;
```

Boolean Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Boolean
   extends System.ValueType
   implements System.IComparable, System.IConvertible
```

Public Methods

```
public final int CompareTo(
   System.Object obj
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public final System.TypeCode GetTypeCode();
public static boolean Parse(
   System.String value
);
public System.String ToString();
public final System.String ToString(
   System.IFormatProvider provider
);
```

Public Fields

```
public static final System.String FalseString;
public static final System.String TrueString;
```

Buffer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Buffer
   extends System.Object
```

Public Methods

```
public static void BlockCopy(
   System.Array src,
   int srcOffset,
   System.Array dst,
   int dstOffset,
   int count
);
public static int ByteLength(
   System.Array array
);
public static ubyte GetByte(
   System.Array array,
   int index
);
public static void SetByte(
   System.Array array,
   int index,
   ubyte value
);
```

Byte Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Byte
   extends System.ValueType
   implements System.IComparable, System.IFormattable, System.IConvertible
```

Public Methods

```
public final int CompareTo(
   System.Object value
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public final System.TypeCode GetTypeCode();
public static ubyte Parse(
   System.String s
);
public static ubyte Parse(
   System.String s,
   System.Globalization.NumberStyles style
);
public static ubyte Parse(
   System.String s,
   System.IFormatProvider provider
);
public static ubyte Parse(
   System.String s,
   System.Globalization.NumberStyles style,
   System.IFormatProvider provider
);
public System.String ToString();
public final System.String ToString(
   System.IFormatProvider provider
);
public System.String ToString(
   System.String format
);
public final System.String ToString(
   System.String format,
   System.IFormatProvider provider
);
```

Public Fields

```
public static ubyte MaxValue;
public static ubyte MinValue;
```

CannotUnloadAppDomainException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class CannotUnloadAppDomainException
   extends System.SystemException
```

Public Constructors

```
public CannotUnloadAppDomainException();
public CannotUnloadAppDomainException(
   System.String message
);
public CannotUnloadAppDomainException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected CannotUnloadAppDomainException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Char Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Char
   extends System.ValueType
   implements System.IComparable, System.IConvertible
```

Public Methods

```
public final int CompareTo(
   System.Object value
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public static double GetNumericValue(
   char c
);
public static double GetNumericValue(
   System.String s,
   int index
);
public final System.TypeCode GetTypeCode();
public static System.Globalization.UnicodeCategory GetUnicodeCategory(
   char c
);
public static System.Globalization.UnicodeCategory GetUnicodeCategory(
   System.String s,
   int index
);
public static boolean IsControl(
   char c
);
public static boolean IsControl(
   System.String s,
   int index
);
public static boolean IsDigit(
   char c
);
public static boolean IsDigit(
   System.String s,
   int index
);
public static boolean IsLetter(
   char c
);
public static boolean IsLetter(
   System.String s,
   int index
);
public static boolean IsLetterOrDigit(
   char c
);
public static boolean IsLetterOrDigit(
   System.String s,
   int index
);
```

```
public static boolean IsLower(
   char c
);
public static boolean IsLower(
   System.String s,
   int index
);
public static boolean IsNumber(
   char c
);
public static boolean IsNumber(
   System.String s,
   int index
);
public static boolean IsPunctuation(
   char c
);
public static boolean IsPunctuation(
   System.String s,
   int index
);
public static boolean IsSeparator(
   char c
);
public static boolean IsSeparator(
   System.String s,
   int index
);
public static boolean IsSurrogate(
   char c
);
public static boolean IsSurrogate(
   System.String s,
   int index
);
public static boolean IsSymbol(
   char c
);
public static boolean IsSymbol(
   System.String s,
   int index
);
public static boolean IsUpper(
   char c
);
public static boolean IsUpper(
   System.String s,
   int index
);
public static boolean IsWhiteSpace(
   char c
);
public static boolean IsWhiteSpace(
   System.String s,
   int index
);
public static char Parse(
   System.String s
);
public static char ToLower(
   char c
);
public static char ToLower(
   char c,
   System.Globalization.CultureInfo culture
);
public System.String ToString();
public static System.String ToString(
```

```
   char c
);
public final System.String ToString(
   System.IFormatProvider provider
);
public static char ToUpper(
   char c
);
public static char ToUpper(
   char c,
   System.Globalization.CultureInfo culture
);
```

Public Fields

```
public static char MaxValue;
public static char MinValue;
```

CharEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class CharEnumerator
   extends System.Object
   implements System.Collections.IEnumerator, System.ICloneable
```

Public Properties

```
public char get_Current();
```

Public Methods

```
public final System.Object Clone();
public final boolean MoveNext();
public final void Reset();
```

CLSCompliantAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class CLSCompliantAttribute
   extends System.Attribute
```

Public Constructors

```
public CLSCompliantAttribute(
   boolean isCompliant
);
```

Public Properties

```
public boolean get_IsCompliant();
```

Console Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Console
   extends System.Object
```

Public Properties

```
public static System.IO.TextWriter get_Error();
public static System.IO.TextReader get_In();
public static System.IO.TextWriter get_Out();
```

Public Methods

```
public static System.IO.Stream OpenStandardError();
public static System.IO.Stream OpenStandardError(
   int bufferSize
);
public static System.IO.Stream OpenStandardInput();
public static System.IO.Stream OpenStandardInput(
   int bufferSize
);
public static System.IO.Stream OpenStandardOutput();
public static System.IO.Stream OpenStandardOutput(
   int bufferSize
);
public static int Read();
public static System.String ReadLine();
public static void SetError(
   System.IO.TextWriter newError
);
public static void SetIn(
   System.IO.TextReader newIn
);
public static void SetOut(
   System.IO.TextWriter newOut
);
public static void Write(
   boolean value
);
public static void Write(
   char value
);
public static void Write(
   char[] buffer
);
public static void Write(
   double value
);
public static void Write(
   int value
);
public static void Write(
   long value
);
public static void Write(
   float value
);
```

```
public static void Write(
   System.Decimal value
);
public static void Write(
   System.Object value
);
public static void Write(
   System.String value
);
public static void Write(
   System.String format,
   System.Object arg0
);
public static void Write(
   System.String format,
   System.Object[] arg
);
public static void Write(
   char[] buffer,
   int index,
   int count
);
public static void Write(
   System.String format,
   System.Object arg0,
   System.Object arg1
);
public static void Write(
   System.String format,
   System.Object arg0,
   System.Object arg1,
   System.Object arg2
);
public static void WriteLine();
public static void WriteLine(
   boolean value
);
public static void WriteLine(
   char value
);
public static void WriteLine(
   char[] buffer
);
public static void WriteLine(
   double value
);
public static void WriteLine(
   int value
);
public static void WriteLine(
   long value
);
public static void WriteLine(
   float value
);
public static void WriteLine(
   System.Decimal value
);
public static void WriteLine(
   System.Object value
);
public static void WriteLine(
   System.String value
);
public static void WriteLine(
   System.String format,
   System.Object arg0
);
```

```
public static void WriteLine(
   System.String format,
   System.Object[] arg
);
public static void WriteLine(
   char[] buffer,
   int index,
   int count
);
public static void WriteLine(
   System.String format,
   System.Object arg0,
   System.Object arg1
);
public static void WriteLine(
   System.String format,
   System.Object arg0,
   System.Object arg1,
   System.Object arg2
);
```

ContextBoundObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class ContextBoundObject
   extends System.MarshalByRefObject
```

Protected Constructors

```
protected ContextBoundObject();
```

ContextMarshalException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ContextMarshalException
   extends System.SystemException
```

Public Constructors

```
public ContextMarshalException();
public ContextMarshalException(
   System.String message
);
public ContextMarshalException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected ContextMarshalException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ContextStaticAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, AllowMultiple = false, Inherit
ed = false) */
public class ContextStaticAttribute
   extends System.Attribute
```

Public Constructors

```
public ContextStaticAttribute();
```

Convert Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Convert
   extends System.Object
```

Public Methods

```
public static System.Object ChangeType(
   System.Object value,
   System.Type conversionType
);
public static System.Object ChangeType(
   System.Object value,
   System.TypeCode typeCode
);
public static System.Object ChangeType(
   System.Object value,
   System.Type conversionType,
   System.IFormatProvider provider
);
public static System.Object ChangeType(
   System.Object value,
   System.TypeCode typeCode,
   System.IFormatProvider provider
);
public static ubyte[] FromBase64CharArray(
   char[] inArray,
   int offset,
   int length
);
public static ubyte[] FromBase64String(
   System.String s
);
public static System.TypeCode GetTypeCode(
   System.Object value
);
public static boolean IsDBNull(
   System.Object value
);
public static int ToBase64CharArray(
   ubyte[] inArray,
   int offsetIn,
   int length,
   char[] outArray,
   int offsetOut
);
public static System.String ToBase64String(
   ubyte[] inArray
);
public static System.String ToBase64String(
   ubyte[] inArray,
   int offset,
   int length
);
public static boolean ToBoolean(
   boolean value
);
public static boolean ToBoolean(
   ubyte value
```

```
);
public static boolean ToBoolean(
   char value
);
public static boolean ToBoolean(
   double value
);
public static boolean ToBoolean(
   short value
);
public static boolean ToBoolean(
   int value
);
public static boolean ToBoolean(
   long value
);
public static boolean ToBoolean(
   float value
);
public static boolean ToBoolean(
   System.DateTime value
);
public static boolean ToBoolean(
   System.Decimal value
);
public static boolean ToBoolean(
   System.Object value
);
public static boolean ToBoolean(
   System.String value
);
public static boolean ToBoolean(
   System.Object value,
   System.IFormatProvider provider
);
public static boolean ToBoolean(
   System.String value,
   System.IFormatProvider provider
);
public static ubyte ToByte(
   boolean value
);
public static ubyte ToByte(
   ubyte value
);
public static ubyte ToByte(
   char value
);
public static ubyte ToByte(
   double value
);
public static ubyte ToByte(
   short value
);
public static ubyte ToByte(
   int value
);
public static ubyte ToByte(
   long value
);
public static ubyte ToByte(
   float value
);
public static ubyte ToByte(
   System.DateTime value
);
public static ubyte ToByte(
   System.Decimal value
```

```
);
public static ubyte ToByte(
   System.Object value
);
public static ubyte ToByte(
   System.String value
);
public static ubyte ToByte(
   System.Object value,
   System.IFormatProvider provider
);
public static ubyte ToByte(
   System.String value,
   int fromBase
);
public static ubyte ToByte(
   System.String value,
   System.IFormatProvider provider
);
public static char ToChar(
   boolean value
);
public static char ToChar(
   ubyte value
);
public static char ToChar(
   char value
);
public static char ToChar(
   double value
);
public static char ToChar(
   short value
);
public static char ToChar(
   int value
);
public static char ToChar(
   long value
);
public static char ToChar(
   float value
);
public static char ToChar(
   System.DateTime value
);
public static char ToChar(
   System.Decimal value
);
public static char ToChar(
   System.Object value
);
public static char ToChar(
   System.String value
);
public static char ToChar(
   System.Object value,
   System.IFormatProvider provider
);
public static char ToChar(
   System.String value,
   System.IFormatProvider provider
);
public static System.DateTime ToDateTime(
   boolean value
);
public static System.DateTime ToDateTime(
   ubyte value
```

```
);
public static System.DateTime ToDateTime(
   char value
);
public static System.DateTime ToDateTime(
   double value
);
public static System.DateTime ToDateTime(
   short value
);
public static System.DateTime ToDateTime(
   int value
);
public static System.DateTime ToDateTime(
   long value
);
public static System.DateTime ToDateTime(
   float value
);
public static System.DateTime ToDateTime(
   System.DateTime value
);
public static System.DateTime ToDateTime(
   System.Decimal value
);
public static System.DateTime ToDateTime(
   System.Object value
);
public static System.DateTime ToDateTime(
   System.String value
);
public static System.DateTime ToDateTime(
   System.Object value,
   System.IFormatProvider provider
);
public static System.DateTime ToDateTime(
   System.String value,
   System.IFormatProvider provider
);
public static System.Decimal ToDecimal(
   boolean value
);
public static System.Decimal ToDecimal(
   ubyte value
);
public static System.Decimal ToDecimal(
   char value
);
public static System.Decimal ToDecimal(
   double value
);
public static System.Decimal ToDecimal(
   short value
);
public static System.Decimal ToDecimal(
   int value
);
public static System.Decimal ToDecimal(
   long value
);
public static System.Decimal ToDecimal(
   float value
);
public static System.Decimal ToDecimal(
   System.DateTime value
);
public static System.Decimal ToDecimal(
   System.Decimal value
```

```
);
public static System.Decimal ToDecimal(
   System.Object value
);
public static System.Decimal ToDecimal(
   System.String value
);
public static System.Decimal ToDecimal(
   System.Object value,
   System.IFormatProvider provider
);
public static System.Decimal ToDecimal(
   System.String value,
   System.IFormatProvider provider
);
public static double ToDouble(
   boolean value
);
public static double ToDouble(
   ubyte value
);
public static double ToDouble(
   char value
);
public static double ToDouble(
   double value
);
public static double ToDouble(
   short value
);
public static double ToDouble(
   int value
);
public static double ToDouble(
   long value
);
public static double ToDouble(
   float value
);
public static double ToDouble(
   System.DateTime value
);
public static double ToDouble(
   System.Decimal value
);
public static double ToDouble(
   System.Object value
);
public static double ToDouble(
   System.String value
);
public static double ToDouble(
   System.Object value,
   System.IFormatProvider provider
);
public static double ToDouble(
   System.String value,
   System.IFormatProvider provider
);
public static short ToInt16(
   boolean value
);
public static short ToInt16(
   ubyte value
);
public static short ToInt16(
   char value
);
```

```
public static short ToInt16(
   double value
);
public static short ToInt16(
   short value
);
public static short ToInt16(
   int value
);
public static short ToInt16(
   long value
);
public static short ToInt16(
   float value
);
public static short ToInt16(
   System.DateTime value
);
public static short ToInt16(
   System.Decimal value
);
public static short ToInt16(
   System.Object value
);
public static short ToInt16(
   System.String value
);
public static short ToInt16(
   System.Object value,
   System.IFormatProvider provider
);
public static short ToInt16(
   System.String value,
   int fromBase
);
public static short ToInt16(
   System.String value,
   System.IFormatProvider provider
);
public static int ToInt32(
   boolean value
);
public static int ToInt32(
   ubyte value
);
public static int ToInt32(
   char value
);
public static int ToInt32(
   double value
);
public static int ToInt32(
   short value
);
public static int ToInt32(
   int value
);
public static int ToInt32(
   long value
);
public static int ToInt32(
   float value
);
public static int ToInt32(
   System.DateTime value
);
public static int ToInt32(
   System.Decimal value
```

```
);
public static int ToInt32(
   System.Object value
);
public static int ToInt32(
   System.String value
);
public static int ToInt32(
   System.Object value,
   System.IFormatProvider provider
);
public static int ToInt32(
   System.String value,
   int fromBase
);
public static int ToInt32(
   System.String value,
   System.IFormatProvider provider
);
public static long ToInt64(
   boolean value
);
public static long ToInt64(
   ubyte value
);
public static long ToInt64(
   char value
);
public static long ToInt64(
   double value
);
public static long ToInt64(
   short value
);
public static long ToInt64(
   int value
);
public static long ToInt64(
   long value
);
public static long ToInt64(
   float value
);
public static long ToInt64(
   System.DateTime value
);
public static long ToInt64(
   System.Decimal value
);
public static long ToInt64(
   System.Object value
);
public static long ToInt64(
   System.String value
);
public static long ToInt64(
   System.Object value,
   System.IFormatProvider provider
);
public static long ToInt64(
   System.String value,
   int fromBase
);
public static long ToInt64(
   System.String value,
   System.IFormatProvider provider
);
public static float ToSingle(
```

```
   boolean value
);
public static float ToSingle(
   ubyte value
);
public static float ToSingle(
   char value
);
public static float ToSingle(
   double value
);
public static float ToSingle(
   short value
);
public static float ToSingle(
   int value
);
public static float ToSingle(
   long value
);
public static float ToSingle(
   float value
);
public static float ToSingle(
   System.DateTime value
);
public static float ToSingle(
   System.Decimal value
);
public static float ToSingle(
   System.Object value
);
public static float ToSingle(
   System.String value
);
public static float ToSingle(
   System.Object value,
   System.IFormatProvider provider
);
public static float ToSingle(
   System.String value,
   System.IFormatProvider provider
);
public static System.String ToString(
   boolean value
);
public static System.String ToString(
   ubyte value
);
public static System.String ToString(
   char value
);
public static System.String ToString(
   double value
);
public static System.String ToString(
   short value
);
public static System.String ToString(
   int value
);
public static System.String ToString(
   long value
);
public static System.String ToString(
   float value
);
public static System.String ToString(
```

```
   System.DateTime value
);
public static System.String ToString(
   System.Decimal value
);
public static System.String ToString(
   System.Object value
);
public static System.String ToString(
   System.String value
);
public static System.String ToString(
   boolean value,
   System.IFormatProvider provider
);
public static System.String ToString(
   ubyte value,
   int toBase
);
public static System.String ToString(
   ubyte value,
   System.IFormatProvider provider
);
public static System.String ToString(
   char value,
   System.IFormatProvider provider
);
public static System.String ToString(
   double value,
   System.IFormatProvider provider
);
public static System.String ToString(
   short value,
   int toBase
);
public static System.String ToString(
   short value,
   System.IFormatProvider provider
);
public static System.String ToString(
   int value,
   int toBase
);
public static System.String ToString(
   int value,
   System.IFormatProvider provider
);
public static System.String ToString(
   long value,
   int toBase
);
public static System.String ToString(
   long value,
   System.IFormatProvider provider
);
public static System.String ToString(
   float value,
   System.IFormatProvider provider
);
public static System.String ToString(
   System.DateTime value,
   System.IFormatProvider provider
);
public static System.String ToString(
   System.Decimal value,
   System.IFormatProvider provider
);
public static System.String ToString(
```

```
   System.Object value,
   System.IFormatProvider provider
);
public static System.String ToString(
   System.String value,
   System.IFormatProvider provider
);
```

Public Fields

```
public static final System.Object DBNull;
```

CrossAppDomainDelegate Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void CrossAppDomainDelegate();
```

DateTime Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DateTime
   extends System.ValueType
   implements System.IComparable, System.IFormattable, System.IConvertible
```

Public Constructors

```
public DateTime(
   long ticks
);
public DateTime(
   int year,
   int month,
   int day
);
public DateTime(
   int year,
   int month,
   int day,
   System.Globalization.Calendar calendar
);
public DateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second
);
public DateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second,
   int millisecond
);
public DateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second,
   System.Globalization.Calendar calendar
);
public DateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second,
   int millisecond,
   System.Globalization.Calendar calendar
);
```

Public Properties

```
public System.DateTime get_Date();
public int get_Day();
public System.DayOfWeek get_DayOfWeek();
public int get_DayOfYear();
public int get_Hour();
public int get_Millisecond();
public int get_Minute();
public int get_Month();
public static System.DateTime get_Now();
public int get_Second();
public long get_Ticks();
public System.TimeSpan get_TimeOfDay();
public static System.DateTime get_Today();
public static System.DateTime get_UtcNow();
public int get_Year();
```

Public Methods

```
public System.DateTime Add(
   System.TimeSpan value
);
public System.DateTime AddDays(
   double value
);
public System.DateTime AddHours(
   double value
);
public System.DateTime AddMilliseconds(
   double value
);
public System.DateTime AddMinutes(
   double value
);
public System.DateTime AddMonths(
   int months
);
public System.DateTime AddSeconds(
   double value
);
public System.DateTime AddTicks(
   long value
);
public System.DateTime AddYears(
   int value
);
public static int Compare(
   System.DateTime t1,
   System.DateTime t2
);
public final int CompareTo(
   System.Object value
);
public static int DaysInMonth(
   int year,
   int month
);
public boolean Equals(
   System.Object value
);
public static boolean Equals(
   System.DateTime t1,
```

```
   System.DateTime t2
);
public static System.DateTime FromFileTime(
   long fileTime
);
public static System.DateTime FromOADate(
   double d
);
public System.String[] GetDateTimeFormats();
public System.String[] GetDateTimeFormats(
   char format
);
public System.String[] GetDateTimeFormats(
   System.IFormatProvider provider
);
public System.String[] GetDateTimeFormats(
   char format,
   System.IFormatProvider provider
);
public int GetHashCode();
public final System.TypeCode GetTypeCode();
public static boolean IsLeapYear(
   int year
);
public static System.DateTime Parse(
   System.String s
);
public static System.DateTime Parse(
   System.String s,
   System.IFormatProvider provider
);
public static System.DateTime Parse(
   System.String s,
   System.IFormatProvider provider,
   System.Globalization.DateTimeStyles styles
);
public static System.DateTime ParseExact(
   System.String s,
   System.String format,
   System.IFormatProvider provider
);
public static System.DateTime ParseExact(
   System.String s,
   System.String format,
   System.IFormatProvider provider,
   System.Globalization.DateTimeStyles style
);
public static System.DateTime ParseExact(
   System.String s,
   System.String[] formats,
   System.IFormatProvider provider,
   System.Globalization.DateTimeStyles style
);
public System.DateTime Subtract(
   System.TimeSpan value
);
public System.TimeSpan Subtract(
   System.DateTime value
);
public long ToFileTime();
public System.DateTime ToLocalTime();
public System.String ToLongDateString();
public System.String ToLongTimeString();
public double ToOADate();
public System.String ToShortDateString();
public System.String ToShortTimeString();
public System.String ToString();
public final System.String ToString(
```

```
   System.IFormatProvider provider
);
public System.String ToString(
   System.String format
);
public final System.String ToString(
   System.String format,
   System.IFormatProvider provider
);
public System.DateTime ToUniversalTime();
```

Public Operators

```
public static System.DateTime op_Addition(
   System.DateTime d,
   System.TimeSpan t
);
public static boolean op_Equality(
   System.DateTime d1,
   System.DateTime d2
);
public static boolean op_GreaterThan(
   System.DateTime t1,
   System.DateTime t2
);
public static boolean op_GreaterThanOrEqual(
   System.DateTime t1,
   System.DateTime t2
);
public static boolean op_Inequality(
   System.DateTime d1,
   System.DateTime d2
);
public static boolean op_LessThan(
   System.DateTime t1,
   System.DateTime t2
);
public static boolean op_LessThanOrEqual(
   System.DateTime t1,
   System.DateTime t2
);
public static System.DateTime op_Subtraction(
   System.DateTime d,
   System.TimeSpan t
);
public static System.TimeSpan op_Subtraction(
   System.DateTime d1,
   System.DateTime d2
);
```

Public Fields

```
public static final System.DateTime MaxValue;
public static final System.DateTime MinValue;
```

DayOfWeek Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum DayOfWeek
```

Members

Friday

Monday

Saturday

Sunday

Thursday

Tuesday

Wednesday

DBNull Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DBNull
   extends System.Object
   implements System.Runtime.Serialization.ISerializable, System.IConvertible
```

Public Methods

```
public final void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public final System.TypeCode GetTypeCode();
public System.String ToString();
public final System.String ToString(
   System.IFormatProvider provider
);
```

Public Fields

```
public static final System.DBNull Value;
```

Decimal Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Decimal
   extends System.ValueType
   implements System.IFormattable, System.IComparable, System.IConvertible
```

Public Constructors

```
public Decimal(
   double value
);
public Decimal(
   int value
);
public Decimal(
   int[] bits
);
public Decimal(
   long value
);
public Decimal(
   float value
);
public Decimal(
   int lo,
   int mid,
   int hi,
   boolean isNegative,
   ubyte scale
);
```

Public Methods

```
public static System.Decimal Add(
   System.Decimal d1,
   System.Decimal d2
);
public static int Compare(
   System.Decimal d1,
   System.Decimal d2
);
public final int CompareTo(
   System.Object value
);
public static System.Decimal Divide(
   System.Decimal d1,
   System.Decimal d2
);
public boolean Equals(
   System.Object value
);
public static boolean Equals(
   System.Decimal d1,
   System.Decimal d2
);
public static System.Decimal Floor(
```

```
   System.Decimal d
);
public static System.Decimal FromOACurrency(
   long cy
);
public static int[] GetBits(
   System.Decimal d
);
public int GetHashCode();
public final System.TypeCode GetTypeCode();
public static System.Decimal Multiply(
   System.Decimal d1,
   System.Decimal d2
);
public static System.Decimal Negate(
   System.Decimal d
);
public static System.Decimal Parse(
   System.String s
);
public static System.Decimal Parse(
   System.String s,
   System.Globalization.NumberStyles style
);
public static System.Decimal Parse(
   System.String s,
   System.IFormatProvider provider
);
public static System.Decimal Parse(
   System.String s,
   System.Globalization.NumberStyles style,
   System.IFormatProvider provider
);
public static System.Decimal Remainder(
   System.Decimal d1,
   System.Decimal d2
);
public static System.Decimal Round(
   System.Decimal d,
   int decimals
);
public static System.Decimal Subtract(
   System.Decimal d1,
   System.Decimal d2
);
public static ubyte ToByte(
   System.Decimal value
);
public static double ToDouble(
   System.Decimal d
);
public static short ToInt16(
   System.Decimal value
);
public static int ToInt32(
   System.Decimal d
);
public static long ToInt64(
   System.Decimal d
);
public static long ToOACurrency(
   System.Decimal value
);
public static float ToSingle(
   System.Decimal d
);
public System.String ToString();
public final System.String ToString(
```

```
   System.IFormatProvider provider
);
public System.String ToString(
   System.String format
);
public final System.String ToString(
   System.String format,
   System.IFormatProvider provider
);
public static System.Decimal Truncate(
   System.Decimal d
);
```

Public Operators

```
public static System.Decimal op_Addition(
   System.Decimal d1,
   System.Decimal d2
);
public static System.Decimal op_Decrement(
   System.Decimal d
);
public static System.Decimal op_Division(
   System.Decimal d1,
   System.Decimal d2
);
public static boolean op_Equality(
   System.Decimal d1,
   System.Decimal d2
);
public static ubyte op_Explicit(
   System.Decimal value
);
public static char op_Explicit(
   System.Decimal value
);
public static double op_Explicit(
   System.Decimal value
);
public static short op_Explicit(
   System.Decimal value
);
public static int op_Explicit(
   System.Decimal value
);
public static long op_Explicit(
   System.Decimal value
);
public static float op_Explicit(
   System.Decimal value
);
public static System.Decimal op_Explicit(
   double value
);
public static System.Decimal op_Explicit(
   float value
);
public static boolean op_GreaterThan(
   System.Decimal d1,
   System.Decimal d2
);
public static boolean op_GreaterThanOrEqual(
   System.Decimal d1,
   System.Decimal d2
);
public static System.Decimal op_Implicit(
```

```
   ubyte value
);
public static System.Decimal op_Implicit(
   char value
);
public static System.Decimal op_Implicit(
   short value
);
public static System.Decimal op_Implicit(
   int value
);
public static System.Decimal op_Implicit(
   long value
);
public static System.Decimal op_Increment(
   System.Decimal d
);
public static boolean op_Inequality(
   System.Decimal d1,
   System.Decimal d2
);
public static boolean op_LessThan(
   System.Decimal d1,
   System.Decimal d2
);
public static boolean op_LessThanOrEqual(
   System.Decimal d1,
   System.Decimal d2
);
public static System.Decimal op_Modulus(
   System.Decimal d1,
   System.Decimal d2
);
public static System.Decimal op_Multiply(
   System.Decimal d1,
   System.Decimal d2
);
public static System.Decimal op_Subtraction(
   System.Decimal d1,
   System.Decimal d2
);
public static System.Decimal op_UnaryNegation(
   System.Decimal d
);
public static System.Decimal op_UnaryPlus(
   System.Decimal d
);
```

Public Fields

```
/** @attribute System.Runtime.CompilerServices.DecimalConstantAttribute(Value = 7922816251426
4337593543950335) */
public static final System.Decimal MaxValue;
/** @attribute System.Runtime.CompilerServices.DecimalConstantAttribute(Value = -1) */
public static final System.Decimal MinusOne;
/** @attribute System.Runtime.CompilerServices.DecimalConstantAttribute(Value = -792281625142
64337593543950335) */
public static final System.Decimal MinValue;
/** @attribute System.Runtime.CompilerServices.DecimalConstantAttribute(Value = 1) */
public static final System.Decimal One;
/** @attribute System.Runtime.CompilerServices.DecimalConstantAttribute(Value = 0) */
public static final System.Decimal Zero;
```

Delegate Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDual) */
public abstract class Delegate
   extends System.Object
   implements System.ICloneable, System.Runtime.Serialization.ISerializable
```

Public Properties

```
public System.Reflection.MethodInfo get_Method();
public System.Object get_Target();
```

Public Methods

```
public System.Object Clone();
public static System.Delegate Combine(
   System.Delegate[] delegates
);
public static System.Delegate Combine(
   System.Delegate a,
   System.Delegate b
);
public static System.Delegate CreateDelegate(
   System.Type type,
   System.Reflection.MethodInfo method
);
public static System.Delegate CreateDelegate(
   System.Type type,
   System.Object target,
   System.String method
);
public static System.Delegate CreateDelegate(
   System.Type type,
   System.Type target,
   System.String method
);
public static System.Delegate CreateDelegate(
   System.Type type,
   System.Object target,
   System.String method,
   boolean ignoreCase
);
public System.Object DynamicInvoke(
   System.Object[] args
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public System.Delegate[] GetInvocationList();
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public static System.Delegate Remove(
   System.Delegate source,
```

```
   System.Delegate value
);
```

Public Operators

```
public static boolean op_Equality(
   System.Delegate d1,
   System.Delegate d2
);
public static boolean op_Inequality(
   System.Delegate d1,
   System.Delegate d2
);
```

Protected Constructors

```
protected Delegate(
   System.Object target,
   System.String method
);
protected Delegate(
   System.Type target,
   System.String method
);
```

Protected Methods

```
protected System.Delegate CombineImpl(
   System.Delegate d
);
protected System.Object DynamicInvokeImpl(
   System.Object[] args
);
protected System.Reflection.MethodInfo GetMethodImpl();
protected System.Delegate RemoveImpl(
   System.Delegate d
);
```

DivideByZeroException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DivideByZeroException
   extends System.ArithmeticException
```

Public Constructors

```
public DivideByZeroException();
public DivideByZeroException(
   System.String message
);
public DivideByZeroException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected DivideByZeroException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

DllNotFoundException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DllNotFoundException
   extends System.TypeLoadException
```

Public Constructors

```
public DllNotFoundException();
public DllNotFoundException(
   System.String message
);
public DllNotFoundException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected DllNotFoundException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Double Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Double
   extends System.ValueType
   implements System.IComparable, System.IFormattable, System.IConvertible
```

Public Methods

```
public final int CompareTo(
   System.Object value
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public final System.TypeCode GetTypeCode();
public static boolean IsInfinity(
   double d
);
public static boolean IsNaN(
   double d
);
public static boolean IsNegativeInfinity(
   double d
);
public static boolean IsPositiveInfinity(
   double d
);
public static double Parse(
   System.String s
);
public static double Parse(
   System.String s,
   System.Globalization.NumberStyles style
);
public static double Parse(
   System.String s,
   System.IFormatProvider provider
);
public static double Parse(
   System.String s,
   System.Globalization.NumberStyles style,
   System.IFormatProvider provider
);
public System.String ToString();
public final System.String ToString(
   System.IFormatProvider provider
);
public System.String ToString(
   System.String format
);
public final System.String ToString(
   System.String format,
   System.IFormatProvider provider
);
public static boolean TryParse(
   System.String s,
   System.Globalization.NumberStyles style,
```

```
    System.IFormatProvider provider,
    double result
);
```

Public Fields

```
public static double Epsilon;
public static double MaxValue;
public static double MinValue;
public static double NaN;
public static double NegativeInfinity;
public static double PositiveInfinity;
```

DuplicateWaitObjectException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DuplicateWaitObjectException
   extends System.ArgumentException
```

Public Constructors

```
public DuplicateWaitObjectException();
public DuplicateWaitObjectException(
   System.String parameterName
);
public DuplicateWaitObjectException(
   System.String parameterName,
   System.String message
);
```

Protected Constructors

```
protected DuplicateWaitObjectException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

EntryPointNotFoundException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class EntryPointNotFoundException
   extends System.TypeLoadException
```

Public Constructors

```
public EntryPointNotFoundException();
public EntryPointNotFoundException(
   System.String message
);
public EntryPointNotFoundException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected EntryPointNotFoundException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Enum Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class Enum
   extends System.ValueType
   implements System.IComparable, System.IFormattable, System.IConvertible
```

Public Methods

```
public final int CompareTo(
   System.Object target
);
public boolean Equals(
   System.Object obj
);
public static System.String Format(
   System.Type enumType,
   System.Object value,
   System.String format
);
public int GetHashCode();
public static System.String GetName(
   System.Type enumType,
   System.Object value
);
public static System.String[] GetNames(
   System.Type enumType
);
public final System.TypeCode GetTypeCode();
public static System.Type GetUnderlyingType(
   System.Type enumType
);
public static System.Array GetValues(
   System.Type enumType
);
public static boolean IsDefined(
   System.Type enumType,
   System.Object value
);
public static System.Object Parse(
   System.Type enumType,
   System.String value
);
public static System.Object Parse(
   System.Type enumType,
   System.String value,
   boolean ignoreCase
);
public static System.Object ToObject(
   System.Type enumType,
   ubyte value
);
public static System.Object ToObject(
   System.Type enumType,
   short value
);
public static System.Object ToObject(
   System.Type enumType,
   int value
```

```
);
public static System.Object ToObject(
   System.Type enumType,
   long value
);
public static System.Object ToObject(
   System.Type enumType,
   System.Object value
);
public System.String ToString();
public final System.String ToString(
   System.IFormatProvider provider
);
public System.String ToString(
   System.String format
);
public final System.String ToString(
   System.String format,
   System.IFormatProvider provider
);
```

Protected Constructors

```
protected Enum();
```

Environment Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Environment
   extends System.Object
```

Public Properties

```
public static System.String get_CommandLine();
public static System.String get_CurrentDirectory();
public static void set_CurrentDirectory(
   System.String value
);
public static int get_ExitCode();
public static void set_ExitCode(
   int value
);
public boolean get_HasShutdownStarted();
public static System.String get_MachineName();
public static System.String get_NewLine();
public static System.OperatingSystem get_OSVersion();
public static System.String get_StackTrace();
public static System.String get_SystemDirectory();
public static int get_TickCount();
public static System.String get_UserDomainName();
public static boolean get_UserInteractive();
public static System.String get_UserName();
public static System.Version get_Version();
public static long get_WorkingSet();
```

Public Methods

```
public static void Exit(
   int exitCode
);
public static System.String ExpandEnvironmentVariables(
   System.String name
);
public static System.String[] GetCommandLineArgs();
public static System.String GetEnvironmentVariable(
   System.String variable
);
public static System.Collections.IDictionary GetEnvironmentVariables();
public static System.String GetFolderPath(
   System.Environment+SpecialFolder folder
);
public static System.String[] GetLogicalDrives();
```

Environment.SpecialFolder Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SpecialFolder
```

Members

ApplicationData

CommonApplicationData

CommonProgramFiles

Cookies

DesktopDirectory

Favorites

History

InternetCache

LocalApplicationData

Personal

ProgramFiles

Programs

Recent

SendTo

StartMenu

Startup

System

Templates

EventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class EventArgs
   extends System.Object
```

Public Constructors

```
public EventArgs();
```

Public Fields

```
public static final System.EventArgs Empty;
```

EventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void EventHandler(
   System.Object sender,
   System.EventArgs e
);
```

Exception Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDual) */
public class Exception
   extends System.Object
   implements System.Runtime.Serialization.ISerializable
```

Public Constructors

```
public Exception();
public Exception(
   System.String message
);
public Exception(
   System.String message,
   System.Exception innerException
);
```

Public Properties

```
public System.String get_HelpLink();
public void set_HelpLink(
   System.String value
);
public System.Exception get_InnerException();
public System.String get_Message();
public System.String get_Source();
public void set_Source(
   System.String value
);
public System.String get_StackTrace();
public System.Reflection.MethodBase get_TargetSite();
```

Public Methods

```
public System.Exception GetBaseException();
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public System.String ToString();
```

Protected Constructors

```
protected Exception(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Properties

```
protected int get_HResult();
protected void set_HResult(
   int value
);
```

ExecutionEngineException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ExecutionEngineException
   extends System.SystemException
```

Public Constructors

```
public ExecutionEngineException();
public ExecutionEngineException(
   System.String message
);
public ExecutionEngineException(
   System.String message,
   System.Exception innerException
);
```

FieldAccessException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class FieldAccessException
   extends System.MemberAccessException
```

Public Constructors

```
public FieldAccessException();
public FieldAccessException(
   System.String message
);
public FieldAccessException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected FieldAccessException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

FlagsAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Enum, AllowMultiple = false, Inherite
d = false) */
public class FlagsAttribute
   extends System.Attribute
```

Public Constructors

```
public FlagsAttribute();
```

FormatException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class FormatException
   extends System.SystemException
```

Public Constructors

```
public FormatException();
public FormatException(
   System.String message
);
public FormatException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected FormatException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

GC Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class GC
   extends System.Object
```

Public Properties

```
public static int get_MaxGeneration();
```

Public Methods

```
public static void Collect();
public static void Collect(
   int generation
);
public static int GetGeneration(
   System.Object obj
);
public static int GetGeneration(
   System.WeakReference wo
);
public static long GetTotalMemory(
   boolean forceFullCollection
);
public static void KeepAlive(
   System.Object obj
);
public static void ReRegisterForFinalize(
   System.Object obj
);
public static void SuppressFinalize(
   System.Object obj
);
public static void WaitForPendingFinalizers();
```

Guid Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Guid
   extends System.ValueType
   implements System.IFormattable, System.IComparable
```

Public Constructors

```
public Guid(
   ubyte[] b
);
public Guid(
   System.String g
);
public Guid(
   int a,
   short b,
   short c,
   ubyte[] d
);
public Guid(
   int a,
   short b,
   short c,
   ubyte d,
   ubyte e,
   ubyte f,
   ubyte g,
   ubyte h,
   ubyte i,
   ubyte j,
   ubyte k
);
```

Public Methods

```
public final int CompareTo(
   System.Object value
);
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public static System.Guid NewGuid();
public ubyte[] ToByteArray();
public System.String ToString();
public System.String ToString(
   System.String format
);
public final System.String ToString(
   System.String format,
   System.IFormatProvider provider
);
```

Public Operators

```
public static boolean op_Equality(
   System.Guid a,
   System.Guid b
);
public static boolean op_Inequality(
   System.Guid a,
   System.Guid b
);
```

Public Fields

```
public static final System.Guid Empty;
```

IAppDomainSetup Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "27FFF232-A7A8-40dd-8D4A-
734AD59FCD41") */
public interface IAppDomainSetup
```

Public Properties

```
System.String get_ApplicationBase();
void set_ApplicationBase(
   System.String value
);
System.String get_ApplicationName();
void set_ApplicationName(
   System.String value
);
System.String get_CachePath();
void set_CachePath(
   System.String value
);
System.String get_ConfigurationFile();
void set_ConfigurationFile(
   System.String value
);
System.String get_DynamicBase();
void set_DynamicBase(
   System.String value
);
System.String get_LicenseFile();
void set_LicenseFile(
   System.String value
);
System.String get_PrivateBinPath();
void set_PrivateBinPath(
   System.String value
);
System.String get_PrivateBinPathProbe();
void set_PrivateBinPathProbe(
   System.String value
);
System.String get_ShadowCopyDirectories();
void set_ShadowCopyDirectories(
   System.String value
);
System.String get_ShadowCopyFiles();
void set_ShadowCopyFiles(
   System.String value
);
```

IAsyncResult Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IAsyncResult
```

Public Properties

```
System.Object get_AsyncState();
System.Threading.WaitHandle get_AsyncWaitHandle();
boolean get_CompletedSynchronously();
boolean get_IsCompleted();
```

ICloneable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICloneable
```

Public Methods

```
System.Object Clone();
```

IComparable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IComparable
```

Public Methods

```
int CompareTo(
   System.Object obj
);
```

IConvertible Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public interface IConvertible
```

Public Methods

```
System.TypeCode GetTypeCode();
boolean ToBoolean(
   System.IFormatProvider provider
);
ubyte ToByte(
   System.IFormatProvider provider
);
char ToChar(
   System.IFormatProvider provider
);
System.DateTime ToDateTime(
   System.IFormatProvider provider
);
System.Decimal ToDecimal(
   System.IFormatProvider provider
);
double ToDouble(
   System.IFormatProvider provider
);
short ToInt16(
   System.IFormatProvider provider
);
int ToInt32(
   System.IFormatProvider provider
);
long ToInt64(
   System.IFormatProvider provider
);
byte ToSByte(
   System.IFormatProvider provider
);
float ToSingle(
   System.IFormatProvider provider
);
System.String ToString(
   System.IFormatProvider provider
);
System.Object ToType(
   System.Type conversionType,
   System.IFormatProvider provider
);
System.UInt16 ToUInt16(
   System.IFormatProvider provider
);
System.UInt32 ToUInt32(
   System.IFormatProvider provider
);
System.UInt64 ToUInt64(
   System.IFormatProvider provider
);
```

ICustomFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICustomFormatter
```

Public Methods

```
System.String Format(
   System.String format,
   System.Object arg,
   System.IFormatProvider formatProvider
);
```

IDisposable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDisposable
```

Public Methods

```
void Dispose();
```

IFormatProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IFormatProvider
```

Public Methods

```
System.Object GetFormat(
   System.Type formatType
);
```

IFormattable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IFormattable
```

Public Methods

```
System.String ToString(
   System.String format,
   System.IFormatProvider formatProvider
);
```

IndexOutOfRangeException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class IndexOutOfRangeException
   extends System.SystemException
```

Public Constructors

```
public IndexOutOfRangeException();
public IndexOutOfRangeException(
   System.String message
);
public IndexOutOfRangeException(
   System.String message,
   System.Exception innerException
);
```

Int16 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Int16
   extends System.ValueType
   implements System.IComparable, System.IFormattable, System.IConvertible
```

Public Methods

```
public final int CompareTo(
   System.Object value
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public final System.TypeCode GetTypeCode();
public static short Parse(
   System.String s
);
public static short Parse(
   System.String s,
   System.Globalization.NumberStyles style
);
public static short Parse(
   System.String s,
   System.IFormatProvider provider
);
public static short Parse(
   System.String s,
   System.Globalization.NumberStyles style,
   System.IFormatProvider provider
);
public System.String ToString();
public final System.String ToString(
   System.IFormatProvider provider
);
public System.String ToString(
   System.String format
);
public final System.String ToString(
   System.String format,
   System.IFormatProvider provider
);
```

Public Fields

```
public static short MaxValue;
public static short MinValue;
```

Int32 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Int32
   extends System.ValueType
   implements System.IComparable, System.IFormattable, System.IConvertible
```

Public Methods

```
public final int CompareTo(
   System.Object value
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public final System.TypeCode GetTypeCode();
public static int Parse(
   System.String s
);
public static int Parse(
   System.String s,
   System.Globalization.NumberStyles style
);
public static int Parse(
   System.String s,
   System.IFormatProvider provider
);
public static int Parse(
   System.String s,
   System.Globalization.NumberStyles style,
   System.IFormatProvider provider
);
public System.String ToString();
public final System.String ToString(
   System.IFormatProvider provider
);
public System.String ToString(
   System.String format
);
public final System.String ToString(
   System.String format,
   System.IFormatProvider provider
);
```

Public Fields

```
public static int MaxValue;
public static int MinValue;
```

Int64 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Int64
   extends System.ValueType
   implements System.IComparable, System.IFormattable, System.IConvertible
```

Public Methods

```
public final int CompareTo(
   System.Object value
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public final System.TypeCode GetTypeCode();
public static long Parse(
   System.String s
);
public static long Parse(
   System.String s,
   System.Globalization.NumberStyles style
);
public static long Parse(
   System.String s,
   System.IFormatProvider provider
);
public static long Parse(
   System.String s,
   System.Globalization.NumberStyles style,
   System.IFormatProvider provider
);
public System.String ToString();
public final System.String ToString(
   System.IFormatProvider provider
);
public System.String ToString(
   System.String format
);
public final System.String ToString(
   System.String format,
   System.IFormatProvider provider
);
```

Public Fields

```
public static long MaxValue;
public static long MinValue;
```

IntPtr Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class IntPtr
   extends System.ValueType
   implements System.Runtime.Serialization.ISerializable
```

Public Constructors

```
public IntPtr(
   int value
);
public IntPtr(
   long value
);
```

Public Properties

```
public static int get_Size();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public int ToInt32();
public long ToInt64();
public System.String ToString();
```

Public Operators

```
public static boolean op_Equality(
   System.IntPtr value1,
   System.IntPtr value2
);
public static int op_Explicit(
   System.IntPtr value
);
public static long op_Explicit(
   System.IntPtr value
);
public static System.IntPtr op_Explicit(
   int value
);
public static System.IntPtr op_Explicit(
   long value
);
public static boolean op_Inequality(
   System.IntPtr value1,
   System.IntPtr value2
);
```

Public Fields

```
public static final System.IntPtr Zero;
```

InvalidCastException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class InvalidCastException
   extends System.SystemException
```

Public Constructors

```
public InvalidCastException();
public InvalidCastException(
   System.String message
);
public InvalidCastException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected InvalidCastException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

InvalidOperationException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class InvalidOperationException
   extends System.SystemException
```

Public Constructors

```
public InvalidOperationException();
public InvalidOperationException(
   System.String message
);
public InvalidOperationException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected InvalidOperationException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

InvalidProgramException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class InvalidProgramException
   extends System.SystemException
```

Public Constructors

```
public InvalidProgramException();
public InvalidProgramException(
   System.String message
);
public InvalidProgramException(
   System.String message,
   System.Exception inner
);
```

IServiceProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public interface IServiceProvider
```

Public Methods

```
System.Object GetService(
   System.Type serviceType
);
```

LoaderOptimization Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum LoaderOptimization
```

Members

MultiDomain

MultiDomainHost

NotSpecified

SingleDomain

LoaderOptimizationAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = true) */
public class LoaderOptimizationAttribute
   extends System.Attribute
```

Public Constructors

```
public LoaderOptimizationAttribute(
   ubyte value
);
public LoaderOptimizationAttribute(
   System.LoaderOptimization value
);
```

Public Properties

```
public System.LoaderOptimization get_Value();
```

LocalDataStoreSlot Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LocalDataStoreSlot
   extends System.Object
```

Protected Methods

```
protected void Finalize();
```

MarshalByRefObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class MarshalByRefObject
   extends System.Object
```

Public Methods

```
public System.Runtime.Remoting.ObjRef CreateObjRef(
   System.Type requestedType
);
public final System.Object GetLifetimeService();
public System.Object InitializeLifetimeService();
```

Protected Constructors

```
protected MarshalByRefObject();
```

Math Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Math
   extends System.Object
```

Public Methods

```
public static double Abs(
   double value
);
public static short Abs(
   short value
);
public static int Abs(
   int value
);
public static long Abs(
   long value
);
public static float Abs(
   float value
);
public static System.Decimal Abs(
   System.Decimal value
);
public static double Acos(
   double d
);
public static double Asin(
   double d
);
public static double Atan(
   double d
);
public static double Atan2(
   double y,
   double x
);
public static double Ceiling(
   double a
);
public static double Cos(
   double d
);
public static double Cosh(
   double value
);
public static double Exp(
   double d
);
public static double Floor(
   double d
);
public static double IEEERemainder(
   double x,
   double y
);
public static double Log(
   double d
```

```
);
public static double Log(
   double a,
   double newBase
);
public static double Log10(
   double d
);
public static ubyte Max(
   ubyte val1,
   ubyte val2
);
public static double Max(
   double val1,
   double val2
);
public static short Max(
   short val1,
   short val2
);
public static int Max(
   int val1,
   int val2
);
public static long Max(
   long val1,
   long val2
);
public static float Max(
   float val1,
   float val2
);
public static System.Decimal Max(
   System.Decimal val1,
   System.Decimal val2
);
public static ubyte Min(
   ubyte val1,
   ubyte val2
);
public static double Min(
   double val1,
   double val2
);
public static short Min(
   short val1,
   short val2
);
public static int Min(
   int val1,
   int val2
);
public static long Min(
   long val1,
   long val2
);
public static float Min(
   float val1,
   float val2
);
public static System.Decimal Min(
   System.Decimal val1,
   System.Decimal val2
);
public static double Pow(
   double x,
   double y
);
```

```
public static double Round(
   double a
);
public static System.Decimal Round(
   System.Decimal d
);
public static double Round(
   double value,
   int digits
);
public static System.Decimal Round(
   System.Decimal d,
   int decimals
);
public static int Sign(
   double value
);
public static int Sign(
   short value
);
public static int Sign(
   int value
);
public static int Sign(
   long value
);
public static int Sign(
   float value
);
public static int Sign(
   System.Decimal value
);
public static double Sin(
   double a
);
public static double Sinh(
   double value
);
public static double Sqrt(
   double d
);
public static double Tan(
   double a
);
public static double Tanh(
   double value
);
```

Public Fields

```
public static double E;
public static double PI;
```

MemberAccessException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class MemberAccessException
   extends System.SystemException
```

Public Constructors

```
public MemberAccessException();
public MemberAccessException(
   System.String message
);
public MemberAccessException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected MemberAccessException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

MethodAccessException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class MethodAccessException
   extends System.MemberAccessException
```

Public Constructors

```
public MethodAccessException();
public MethodAccessException(
   System.String message
);
public MethodAccessException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected MethodAccessException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

MissingFieldException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class MissingFieldException
   extends System.MissingMemberException
```

Public Constructors

```
public MissingFieldException();
public MissingFieldException(
   System.String message
);
public MissingFieldException(
   System.String message,
   System.Exception inner
);
public MissingFieldException(
   System.String className,
   System.String fieldName
);
```

Public Properties

```
public System.String get_Message();
```

Protected Constructors

```
protected MissingFieldException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

MissingMemberException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class MissingMemberException
   extends System.MemberAccessException
```

Public Constructors

```
public MissingMemberException();
public MissingMemberException(
   System.String message
);
public MissingMemberException(
   System.String message,
   System.Exception inner
);
public MissingMemberException(
   System.String className,
   System.String memberName
);
```

Public Properties

```
public System.String get_Message();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected MissingMemberException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Fields

```
protected System.String ClassName;
protected System.String MemberName;
protected ubyte[] Signature;
```

MissingMethodException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class MissingMethodException
   extends System.MissingMemberException
```

Public Constructors

```
public MissingMethodException();
public MissingMethodException(
   System.String message
);
public MissingMethodException(
   System.String message,
   System.Exception inner
);
public MissingMethodException(
   System.String className,
   System.String methodName
);
```

Public Properties

```
public System.String get_Message();
```

Protected Constructors

```
protected MissingMethodException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

MTAThreadAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = true) */
public class MTAThreadAttribute
   extends System.Attribute
```

Public Constructors

```
public MTAThreadAttribute();
```

MulticastDelegate Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class MulticastDelegate
   extends System.Delegate
```

Public Methods

```
public final boolean Equals(
   System.Object obj
);
public final int GetHashCode();
public final System.Delegate[] GetInvocationList();
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Public Operators

```
public static boolean op_Equality(
   System.MulticastDelegate d1,
   System.MulticastDelegate d2
);
public static boolean op_Inequality(
   System.MulticastDelegate d1,
   System.MulticastDelegate d2
);
```

Protected Constructors

```
protected MulticastDelegate(
   System.Object target,
   System.String method
);
protected MulticastDelegate(
   System.Type target,
   System.String method
);
```

Protected Methods

```
protected final System.Delegate CombineImpl(
   System.Delegate follow
);
protected final System.Object DynamicInvokeImpl(
   System.Object[] args
);
protected final System.Delegate RemoveImpl(
   System.Delegate value
);
```

MulticastNotSupportedException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class MulticastNotSupportedException
   extends System.SystemException
```

Public Constructors

```
public MulticastNotSupportedException();
public MulticastNotSupportedException(
   System.String message
);
public MulticastNotSupportedException(
   System.String message,
   System.Exception inner
);
```

NonSerializedAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, AllowMultiple = false, Inherit
ed = true) */
public class NonSerializedAttribute
   extends System.Attribute
```

Public Constructors

```
public NonSerializedAttribute();
```

NotFiniteNumberException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class NotFiniteNumberException
   extends System.ArithmeticException
```

Public Constructors

```
public NotFiniteNumberException();
public NotFiniteNumberException(
   double offendingNumber
);
public NotFiniteNumberException(
   System.String message
);
public NotFiniteNumberException(
   System.String message,
   double offendingNumber
);
public NotFiniteNumberException(
   System.String message,
   double offendingNumber,
   System.Exception innerException
);
```

Public Properties

```
public double get_OffendingNumber();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected NotFiniteNumberException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

NotImplementedException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class NotImplementedException
   extends System.SystemException
```

Public Constructors

```
public NotImplementedException();
public NotImplementedException(
   System.String message
);
public NotImplementedException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected NotImplementedException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

NotSupportedException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class NotSupportedException
   extends System.SystemException
```

Public Constructors

```
public NotSupportedException();
public NotSupportedException(
   System.String message
);
public NotSupportedException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected NotSupportedException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

NullReferenceException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class NullReferenceException
   extends System.SystemException
```

Public Constructors

```
public NullReferenceException();
public NullReferenceException(
   System.String message
);
public NullReferenceException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected NullReferenceException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Object Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDual) */
public class Object
```

Public Constructors

```
public Object();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public static boolean Equals(
   System.Object objA,
   System.Object objB
);
public int GetHashCode();
public System.Type GetType();
public static boolean ReferenceEquals(
   System.Object objA,
   System.Object objB
);
public System.String ToString();
```

Protected Methods

```
protected void Finalize();
protected System.Object MemberwiseClone();
```

ObjectDisposedException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ObjectDisposedException
   extends System.InvalidOperationException
```

Public Constructors

```
public ObjectDisposedException(
   System.String objectName
);
public ObjectDisposedException(
   System.String objectName,
   System.String message
);
```

Public Properties

```
public System.String get_Message();
public System.String get_ObjectName();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected ObjectDisposedException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ObsoleteAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, Enum, Constructor, Met
hod, Property, Field, Event, Interface, Delegate, AllowMultiple = false, Inherited = false) *
/
public class ObsoleteAttribute
   extends System.Attribute
```

Public Constructors

```
public ObsoleteAttribute();
public ObsoleteAttribute(
   System.String message
);
public ObsoleteAttribute(
   System.String message,
   boolean error
);
```

Public Properties

```
public boolean get_IsError();
public System.String get_Message();
```

OperatingSystem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class OperatingSystem
   extends System.Object
   implements System.ICloneable
```

Public Constructors

```
public OperatingSystem(
   System.PlatformID platform,
   System.Version version
);
```

Public Properties

```
public System.PlatformID get_Platform();
public System.Version get_Version();
```

Public Methods

```
public final System.Object Clone();
public System.String ToString();
```

OutOfMemoryException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class OutOfMemoryException
   extends System.SystemException
```

Public Constructors

```
public OutOfMemoryException();
public OutOfMemoryException(
   System.String message
);
public OutOfMemoryException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected OutOfMemoryException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

OverflowException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class OverflowException
   extends System.ArithmeticException
```

Public Constructors

```
public OverflowException();
public OverflowException(
   System.String message
);
public OverflowException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected OverflowException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ParamArrayAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Parameter, AllowMultiple = false, Inh
erited = true) */
public class ParamArrayAttribute
   extends System.Attribute
```

Public Constructors

```
public ParamArrayAttribute();
```

PlatformID Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PlatformID
```

Members

Win32NT

Win32S

Win32Windows

PlatformNotSupportedException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class PlatformNotSupportedException
   extends System.NotSupportedException
```

Public Constructors

```
public PlatformNotSupportedException();
public PlatformNotSupportedException(
   System.String message
);
public PlatformNotSupportedException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected PlatformNotSupportedException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Random Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Random
   extends System.Object
```

Public Constructors

```
public Random();
public Random(
   int Seed
);
```

Public Methods

```
public int Next();
public int Next(
   int maxValue
);
public int Next(
   int minValue,
   int maxValue
);
public void NextBytes(
   ubyte[] buffer
);
public double NextDouble();
```

Protected Methods

```
protected double Sample();
```

RankException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RankException
   extends System.SystemException
```

Public Constructors

```
public RankException();
public RankException(
   System.String message
);
public RankException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected RankException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ResolveEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ResolveEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ResolveEventArgs(
   System.String name
);
```

Public Properties

```
public System.String get_Name();
```

ResolveEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
System.Reflection.Assembly ResolveEventHandler(
   System.Object sender,
   System.ResolveEventArgs args
);
```

RuntimeArgumentHandle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RuntimeArgumentHandle
   extends System.ValueType
```

RuntimeFieldHandle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RuntimeFieldHandle
   extends System.ValueType
   implements System.Runtime.Serialization.ISerializable
```

Public Properties

```
public System.IntPtr get_Value();
```

Public Methods

```
public final void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

RuntimeMethodHandle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RuntimeMethodHandle
   extends System.ValueType
   implements System.Runtime.Serialization.ISerializable
```

Public Properties

```
public System.IntPtr get_Value();
```

Public Methods

```
public System.IntPtr GetFunctionPointer();
public final void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

RuntimeTypeHandle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RuntimeTypeHandle
   extends System.ValueType
   implements System.Runtime.Serialization.ISerializable
```

Public Properties

```
public System.IntPtr get_Value();
```

Public Methods

```
public final void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

SByte Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public class SByte
   extends System.ValueType
   implements System.IComparable, System.IFormattable, System.IConvertible
```

Public Methods

```
public final int CompareTo(
   System.Object obj
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public final System.TypeCode GetTypeCode();
public System.String ToString();
public final System.String ToString(
   System.IFormatProvider provider
);
public System.String ToString(
   System.String format
);
public final System.String ToString(
   System.String format,
   System.IFormatProvider provider
);
```

Public Fields

```
public static byte MaxValue;
public static byte MinValue;
```

SerializableAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, Enum, Delegate, AllowM
ultiple = false, Inherited = false) */
public class SerializableAttribute
   extends System.Attribute
```

Public Constructors

```
public SerializableAttribute();
```

Single Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Single
   extends System.ValueType
   implements System.IComparable, System.IFormattable, System.IConvertible
```

Public Methods

```
public final int CompareTo(
   System.Object value
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public final System.TypeCode GetTypeCode();
public static boolean IsInfinity(
   float f
);
public static boolean IsNaN(
   float f
);
public static boolean IsNegativeInfinity(
   float f
);
public static boolean IsPositiveInfinity(
   float f
);
public static float Parse(
   System.String s
);
public static float Parse(
   System.String s,
   System.Globalization.NumberStyles style
);
public static float Parse(
   System.String s,
   System.IFormatProvider provider
);
public static float Parse(
   System.String s,
   System.Globalization.NumberStyles style,
   System.IFormatProvider provider
);
public System.String ToString();
public final System.String ToString(
   System.IFormatProvider provider
);
public System.String ToString(
   System.String format
);
public final System.String ToString(
   System.String format,
   System.IFormatProvider provider
);
```

Public Fields

```
public static float Epsilon;
public static float MaxValue;
public static float MinValue;
public static float NaN;
public static float NegativeInfinity;
public static float PositiveInfinity;
```

StackOverflowException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StackOverflowException
   extends System.SystemException
```

Public Constructors

```
public StackOverflowException();
public StackOverflowException(
   System.String message
);
public StackOverflowException(
   System.String message,
   System.Exception innerException
);
```

STAThreadAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = true) */
public class STAThreadAttribute
   extends System.Attribute
```

Public Constructors

```
public STAThreadAttribute();
```

String Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Chars") */
public class String
   extends System.Object
   implements System.IComparable, System.ICloneable, System.IConvertible, System.Collections.
IEnumerable
```

Public Constructors

```
public String(
   char[] value
);
public String(
   char c,
   int count
);
public String(
   char[] value,
   int startIndex,
   int length
);
```

Public Properties

```
public char get_Chars(
   int index
);
public int get_Length();
```

Public Methods

```
public final System.Object Clone();
public static int Compare(
   System.String strA,
   System.String strB
);
public static int Compare(
   System.String strA,
   System.String strB,
   boolean ignoreCase
);
public static int Compare(
   System.String strA,
   System.String strB,
   boolean ignoreCase,
   System.Globalization.CultureInfo culture
);
public static int Compare(
   System.String strA,
   int indexA,
   System.String strB,
   int indexB,
   int length
```

```
);
public static int Compare(
   System.String strA,
   int indexA,
   System.String strB,
   int indexB,
   int length,
   boolean ignoreCase
);
public static int Compare(
   System.String strA,
   int indexA,
   System.String strB,
   int indexB,
   int length,
   boolean ignoreCase,
   System.Globalization.CultureInfo culture
);
public static int CompareOrdinal(
   System.String strA,
   System.String strB
);
public static int CompareOrdinal(
   System.String strA,
   int indexA,
   System.String strB,
   int indexB,
   int length
);
public final int CompareTo(
   System.Object value
);
public int CompareTo(
   System.String strB
);
public static System.String Concat(
   System.Object arg0
);
public static System.String Concat(
   System.Object[] args
);
public static System.String Concat(
   System.String[] values
);
public static System.String Concat(
   System.Object arg0,
   System.Object arg1
);
public static System.String Concat(
   System.String str0,
   System.String str1
);
public static System.String Concat(
   System.Object arg0,
   System.Object arg1,
   System.Object arg2
);
public static System.String Concat(
   System.String str0,
   System.String str1,
   System.String str2
);
public static System.String Concat(
   System.String str0,
   System.String str1,
   System.String str2,
   System.String str3
);
```

```
public static System.String Copy(
   System.String str
);
public void CopyTo(
   int sourceIndex,
   char[] destination,
   int destinationIndex,
   int count
);
public boolean EndsWith(
   System.String value
);
public boolean Equals(
   System.Object obj
);
public boolean Equals(
   System.String value
);
public static boolean Equals(
   System.String a,
   System.String b
);
public static System.String Format(
   System.String format,
   System.Object arg0
);
public static System.String Format(
   System.String format,
   System.Object[] args
);
public static System.String Format(
   System.IFormatProvider provider,
   System.String format,
   System.Object[] args
);
public static System.String Format(
   System.String format,
   System.Object arg0,
   System.Object arg1
);
public static System.String Format(
   System.String format,
   System.Object arg0,
   System.Object arg1,
   System.Object arg2
);
public System.CharEnumerator GetEnumerator();
public int GetHashCode();
public final System.TypeCode GetTypeCode();
public int IndexOf(
   char value
);
public int IndexOf(
   System.String value
);
public int IndexOf(
   char value,
   int startIndex
);
public int IndexOf(
   System.String value,
   int startIndex
);
public int IndexOf(
   char value,
   int startIndex,
   int count
);
```

```
public int IndexOf(
   System.String value,
   int startIndex,
   int count
);
public int IndexOfAny(
   char[] anyOf
);
public int IndexOfAny(
   char[] anyOf,
   int startIndex
);
public int IndexOfAny(
   char[] anyOf,
   int startIndex,
   int count
);
public System.String Insert(
   int startIndex,
   System.String value
);
public static System.String Intern(
   System.String str
);
public static System.String IsInterned(
   System.String str
);
public static System.String Join(
   System.String separator,
   System.String[] value
);
public static System.String Join(
   System.String separator,
   System.String[] value,
   int startIndex,
   int count
);
public int LastIndexOf(
   char value
);
public int LastIndexOf(
   System.String value
);
public int LastIndexOf(
   char value,
   int startIndex
);
public int LastIndexOf(
   System.String value,
   int startIndex
);
public int LastIndexOf(
   char value,
   int startIndex,
   int count
);
public int LastIndexOf(
   System.String value,
   int startIndex,
   int count
);
public int LastIndexOfAny(
   char[] anyOf
);
public int LastIndexOfAny(
   char[] anyOf,
   int startIndex
);
```

```
public int LastIndexOfAny(
   char[] anyOf,
   int startIndex,
   int count
);
public System.String PadLeft(
   int totalWidth
);
public System.String PadLeft(
   int totalWidth,
   char paddingChar
);
public System.String PadRight(
   int totalWidth
);
public System.String PadRight(
   int totalWidth,
   char paddingChar
);
public System.String Remove(
   int startIndex,
   int count
);
public System.String Replace(
   char oldChar,
   char newChar
);
public System.String Replace(
   System.String oldValue,
   System.String newValue
);
public System.String[] Split(
   char[] separator
);
public System.String[] Split(
   char[] separator,
   int count
);
public boolean StartsWith(
   System.String value
);
public System.String Substring(
   int startIndex
);
public System.String Substring(
   int startIndex,
   int length
);
public char[] ToCharArray();
public char[] ToCharArray(
   int startIndex,
   int length
);
public System.String ToLower();
public System.String ToLower(
   System.Globalization.CultureInfo culture
);
public System.String ToString();
public final System.String ToString(
   System.IFormatProvider provider
);
public System.String ToUpper();
public System.String ToUpper(
   System.Globalization.CultureInfo culture
);
public System.String Trim();
public System.String Trim(
   char[] trimChars
```

```
);
public System.String TrimEnd(
   char[] trimChars
);
public System.String TrimStart(
   char[] trimChars
);
```

Public Operators

```
public static boolean op_Equality(
   System.String a,
   System.String b
);
public static boolean op_Inequality(
   System.String a,
   System.String b
);
```

Public Fields

```
public static final System.String Empty;
```

SystemException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SystemException
   extends System.Exception
```

Public Constructors

```
public SystemException();
public SystemException(
   System.String message
);
public SystemException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected SystemException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ThreadStaticAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, AllowMultiple = false, Inherit
ed = false) */
public class ThreadStaticAttribute
   extends System.Attribute
```

Public Constructors

```
public ThreadStaticAttribute();
```

TimeSpan Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class TimeSpan
   extends System.ValueType
   implements System.IComparable
```

Public Constructors

```
public TimeSpan(
   long ticks
);
public TimeSpan(
   int hours,
   int minutes,
   int seconds
);
public TimeSpan(
   int days,
   int hours,
   int minutes,
   int seconds
);
public TimeSpan(
   int days,
   int hours,
   int minutes,
   int seconds,
   int milliseconds
);
```

Public Properties

```
public int get_Days();
public int get_Hours();
public int get_Milliseconds();
public int get_Minutes();
public int get_Seconds();
public long get_Ticks();
public double get_TotalDays();
public double get_TotalHours();
public double get_TotalMilliseconds();
public double get_TotalMinutes();
public double get_TotalSeconds();
```

Public Methods

```
public System.TimeSpan Add(
   System.TimeSpan ts
);
public static int Compare(
   System.TimeSpan t1,
   System.TimeSpan t2
);
public final int CompareTo(
```

```
   System.Object value
);
public System.TimeSpan Duration();
public boolean Equals(
   System.Object value
);
public static boolean Equals(
   System.TimeSpan t1,
   System.TimeSpan t2
);
public static System.TimeSpan FromDays(
   double value
);
public static System.TimeSpan FromHours(
   double value
);
public static System.TimeSpan FromMilliseconds(
   double value
);
public static System.TimeSpan FromMinutes(
   double value
);
public static System.TimeSpan FromSeconds(
   double value
);
public static System.TimeSpan FromTicks(
   long value
);
public int GetHashCode();
public System.TimeSpan Negate();
public static System.TimeSpan Parse(
   System.String s
);
public System.TimeSpan Subtract(
   System.TimeSpan ts
);
public System.String ToString();
```

Public Operators

```
public static System.TimeSpan op_Addition(
   System.TimeSpan t1,
   System.TimeSpan t2
);
public static boolean op_Equality(
   System.TimeSpan t1,
   System.TimeSpan t2
);
public static boolean op_GreaterThan(
   System.TimeSpan t1,
   System.TimeSpan t2
);
public static boolean op_GreaterThanOrEqual(
   System.TimeSpan t1,
   System.TimeSpan t2
);
public static boolean op_Inequality(
   System.TimeSpan t1,
   System.TimeSpan t2
);
public static boolean op_LessThan(
   System.TimeSpan t1,
   System.TimeSpan t2
);
public static boolean op_LessThanOrEqual(
   System.TimeSpan t1,
```

```
   System.TimeSpan t2
);
public static System.TimeSpan op_Subtraction(
   System.TimeSpan t1,
   System.TimeSpan t2
);
public static System.TimeSpan op_UnaryNegation(
   System.TimeSpan t
);
public static System.TimeSpan op_UnaryPlus(
   System.TimeSpan t
);
```

Public Fields

```
public static final System.TimeSpan MaxValue;
public static final System.TimeSpan MinValue;
public static long TicksPerDay;
public static long TicksPerHour;
public static long TicksPerMillisecond;
public static long TicksPerMinute;
public static long TicksPerSecond;
public static final System.TimeSpan Zero;
```

TimeZone Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class TimeZone
   extends System.Object
```

Public Properties

```
public static System.TimeZone get_CurrentTimeZone();
public abstract System.String get_DaylightName();
public abstract System.String get_StandardName();
```

Public Methods

```
public abstract System.Globalization.DaylightTime GetDaylightChanges(
   int year
);
public abstract System.TimeSpan GetUtcOffset(
   System.DateTime time
);
public boolean IsDaylightSavingTime(
   System.DateTime time
);
public static boolean IsDaylightSavingTime(
   System.DateTime time,
   System.Globalization.DaylightTime daylightTimes
);
public System.DateTime ToLocalTime(
   System.DateTime time
);
public System.DateTime ToUniversalTime(
   System.DateTime time
);
```

Protected Constructors

```
protected TimeZone();
```

Type Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDual) */
public abstract class Type
   extends System.Reflection.MemberInfo
   implements System.Reflection.IReflect
```

Public Properties

```
public abstract System.Reflection.Assembly get_Assembly();
public abstract System.String get_AssemblyQualifiedName();
public System.Reflection.TypeAttributes get_Attributes();
public abstract System.Type get_BaseType();
public System.Type get_DeclaringType();
public static System.Reflection.Binder get_DefaultBinder();
public abstract System.String get_FullName();
public abstract System.Guid get_GUID();
public boolean get_HasElementType();
public boolean get_IsAbstract();
public boolean get_IsAnsiClass();
public boolean get_IsArray();
public boolean get_IsAutoClass();
public boolean get_IsAutoLayout();
public boolean get_IsByRef();
public boolean get_IsClass();
public boolean get_IsCOMObject();
public boolean get_IsContextful();
public boolean get_IsEnum();
public boolean get_IsExplicitLayout();
public boolean get_IsImport();
public boolean get_IsInterface();
public boolean get_IsLayoutSequential();
public boolean get_IsMarshalByRef();
public boolean get_IsNestedAssembly();
public boolean get_IsNestedFamANDAssem();
public boolean get_IsNestedFamily();
public boolean get_IsNestedFamORAssem();
public boolean get_IsNestedPrivate();
public boolean get_IsNestedPublic();
public boolean get_IsNotPublic();
public boolean get_IsPointer();
public boolean get_IsPrimitive();
public boolean get_IsPublic();
public boolean get_IsSealed();
public boolean get_IsSerializable();
public boolean get_IsSpecialName();
public boolean get_IsUnicodeClass();
public boolean get_IsValueType();
public System.Reflection.MemberTypes get_MemberType();
public abstract System.Reflection.Module get_Module();
public abstract System.String get_Namespace();
public System.Type get_ReflectedType();
public abstract System.RuntimeTypeHandle get_TypeHandle();
public System.Reflection.ConstructorInfo get_TypeInitializer();
public abstract System.Type get_UnderlyingSystemType();
```

Public Methods

```
public boolean Equals(
   System.Object o
);
public boolean Equals(
   System.Type o
);
public System.Type[] FindInterfaces(
   System.Reflection.TypeFilter filter,
   System.Object filterCriteria
);
public System.Reflection.MemberInfo[] FindMembers(
   System.Reflection.MemberTypes memberType,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.MemberFilter filter,
   System.Object filterCriteria
);
public int GetArrayRank();
public System.Reflection.ConstructorInfo GetConstructor(
   System.Type[] types
);
public System.Reflection.ConstructorInfo GetConstructor(
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
public System.Reflection.ConstructorInfo GetConstructor(
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Reflection.CallingConventions callConvention,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
public System.Reflection.ConstructorInfo[] GetConstructors();
public abstract System.Reflection.ConstructorInfo[] GetConstructors(
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.MemberInfo[] GetDefaultMembers();
public abstract System.Type GetElementType();
public System.Reflection.EventInfo GetEvent(
   System.String name
);
public abstract System.Reflection.EventInfo GetEvent(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.EventInfo[] GetEvents();
public abstract System.Reflection.EventInfo[] GetEvents(
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.FieldInfo GetField(
   System.String name
);
public abstract System.Reflection.FieldInfo GetField(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.FieldInfo[] GetFields();
public abstract System.Reflection.FieldInfo[] GetFields(
   System.Reflection.BindingFlags bindingAttr
);
public int GetHashCode();
public System.Type GetInterface(
   System.String name
);
```

```
public abstract System.Type GetInterface(
   System.String name,
   boolean ignoreCase
);
public System.Reflection.InterfaceMapping GetInterfaceMap(
   System.Type interfaceType
);
public abstract System.Type[] GetInterfaces();
public System.Reflection.MemberInfo[] GetMember(
   System.String name
);
public System.Reflection.MemberInfo[] GetMember(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.MemberInfo[] GetMember(
   System.String name,
   System.Reflection.MemberTypes type,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.MemberInfo[] GetMembers();
public abstract System.Reflection.MemberInfo[] GetMembers(
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.MethodInfo GetMethod(
   System.String name
);
public final System.Reflection.MethodInfo GetMethod(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.MethodInfo GetMethod(
   System.String name,
   System.Type[] types
);
public System.Reflection.MethodInfo GetMethod(
   System.String name,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
public final System.Reflection.MethodInfo GetMethod(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
public System.Reflection.MethodInfo GetMethod(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Reflection.CallingConventions callConvention,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
public System.Reflection.MethodInfo[] GetMethods();
public abstract System.Reflection.MethodInfo[] GetMethods(
   System.Reflection.BindingFlags bindingAttr
);
public System.Type GetNestedType(
   System.String name
);
public abstract System.Type GetNestedType(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Type[] GetNestedTypes();
public abstract System.Type[] GetNestedTypes(
```

```
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.PropertyInfo[] GetProperties();
public abstract System.Reflection.PropertyInfo[] GetProperties(
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.PropertyInfo GetProperty(
   System.String name
);
public final System.Reflection.PropertyInfo GetProperty(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.PropertyInfo GetProperty(
   System.String name,
   System.Type returnType
);
public System.Reflection.PropertyInfo GetProperty(
   System.String name,
   System.Type[] types
);
public System.Reflection.PropertyInfo GetProperty(
   System.String name,
   System.Type returnType,
   System.Type[] types
);
public System.Reflection.PropertyInfo GetProperty(
   System.String name,
   System.Type returnType,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
public final System.Reflection.PropertyInfo GetProperty(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Type returnType,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
public static System.Type GetType(
   System.String typeName
);
public static System.Type GetType(
   System.String typeName,
   boolean throwOnError
);
public static System.Type GetType(
   System.String typeName,
   boolean throwOnError,
   boolean ignoreCase
);
public static System.Type[] GetTypeArray(
   System.Object[] args
);
public static System.TypeCode GetTypeCode(
   System.Type type
);
public static System.Type GetTypeFromCLSID(
   System.Guid clsid
);
public static System.Type GetTypeFromCLSID(
   System.Guid clsid,
   boolean throwOnError
);
public static System.Type GetTypeFromCLSID(
   System.Guid clsid,
   System.String server
```

```
);
public static System.Type GetTypeFromCLSID(
   System.Guid clsid,
   System.String server,
   boolean throwOnError
);
public static System.Type GetTypeFromHandle(
   System.RuntimeTypeHandle handle
);
public static System.Type GetTypeFromProgID(
   System.String progID
);
public static System.Type GetTypeFromProgID(
   System.String progID,
   boolean throwOnError
);
public static System.Type GetTypeFromProgID(
   System.String progID,
   System.String server
);
public static System.Type GetTypeFromProgID(
   System.String progID,
   System.String server,
   boolean throwOnError
);
public static System.RuntimeTypeHandle GetTypeHandle(
   System.Object o
);
/** @attribute System.Diagnostics.DebuggerStepThroughAttribute() */
/** @attribute System.Diagnostics.DebuggerHiddenAttribute() */
public System.Object InvokeMember(
   System.String name,
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Object target,
   System.Object[] args
);
/** @attribute System.Diagnostics.DebuggerHiddenAttribute() */
/** @attribute System.Diagnostics.DebuggerStepThroughAttribute() */
public System.Object InvokeMember(
   System.String name,
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Object target,
   System.Object[] args,
   System.Globalization.CultureInfo culture
);
public abstract System.Object InvokeMember(
   System.String name,
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Object target,
   System.Object[] args,
   System.Reflection.ParameterModifier[] modifiers,
   System.Globalization.CultureInfo culture,
   System.String[] namedParameters
);
public boolean IsAssignableFrom(
   System.Type c
);
public boolean IsInstanceOfType(
   System.Object o
);
public boolean IsSubclassOf(
   System.Type c
);
public System.String ToString();
```

Public Fields

```
public static final char Delimiter;
public static final System.Type[] EmptyTypes;
public static final System.Reflection.MemberFilter FilterAttribute;
public static final System.Reflection.MemberFilter FilterName;
public static final System.Reflection.MemberFilter FilterNameIgnoreCase;
public static final System.Object Missing;
```

Protected Constructors

```
protected Type();
```

Protected Methods

```
protected abstract System.Reflection.TypeAttributes GetAttributeFlagsImpl();
protected abstract System.Reflection.ConstructorInfo GetConstructorImpl(
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Reflection.CallingConventions callConvention,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
protected abstract System.Reflection.MethodInfo GetMethodImpl(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Reflection.CallingConventions callConvention,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
protected abstract System.Reflection.PropertyInfo GetPropertyImpl(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Type returnType,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
protected abstract boolean HasElementTypeImpl();
protected abstract boolean IsArrayImpl();
protected abstract boolean IsByRefImpl();
protected abstract boolean IsCOMObjectImpl();
protected boolean IsContextfulImpl();
protected boolean IsMarshalByRefImpl();
protected abstract boolean IsPointerImpl();
protected abstract boolean IsPrimitiveImpl();
protected boolean IsValueTypeImpl();
```

TypeCode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TypeCode
```

Members

Boolean

Byte

Char

DateTime

DBNull

Decimal

Double

Empty

Int16

Int32

Int64

Object

SByte

Single

String

UInt16

UInt32

UInt64

TypedReference Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public class TypedReference
   extends System.ValueType
```

Public Methods

```
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public static System.Type GetTargetType(
   System.TypedReference value
);
public static System.RuntimeTypeHandle TargetTypeToken(
   System.TypedReference value
);
public static System.Object ToObject(
   System.TypedReference value
);
```

TypeInitializationException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class TypeInitializationException
   extends System.SystemException
```

Public Constructors

```
public TypeInitializationException(
   System.String fullTypeName,
   System.Exception innerException
);
```

Public Properties

```
public System.String get_TypeName();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

TypeLoadException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class TypeLoadException
   extends System.SystemException
```

Public Constructors

```
public TypeLoadException();
public TypeLoadException(
   System.String message
);
public TypeLoadException(
   System.String message,
   System.Exception inner
);
```

Public Properties

```
public System.String get_Message();
public System.String get_TypeName();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected TypeLoadException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

TypeUnloadedException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class TypeUnloadedException
   extends System.SystemException
```

Public Constructors

```
public TypeUnloadedException();
public TypeUnloadedException(
   System.String message
);
public TypeUnloadedException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected TypeUnloadedException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

UInt16 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public class UInt16
   extends System.ValueType
   implements System.IComparable, System.IFormattable, System.IConvertible
```

Public Methods

```
public final int CompareTo(
   System.Object value
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public final System.TypeCode GetTypeCode();
public System.String ToString();
public final System.String ToString(
   System.IFormatProvider provider
);
public System.String ToString(
   System.String format
);
public final System.String ToString(
   System.String format,
   System.IFormatProvider provider
);
```

Public Fields

```
public static System.UInt16 MaxValue;
public static System.UInt16 MinValue;
```

UInt32 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public class UInt32
   extends System.ValueType
   implements System.IComparable, System.IFormattable, System.IConvertible
```

Public Methods

```
public final int CompareTo(
   System.Object value
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public final System.TypeCode GetTypeCode();
public System.String ToString();
public final System.String ToString(
   System.IFormatProvider provider
);
public System.String ToString(
   System.String format
);
public final System.String ToString(
   System.String format,
   System.IFormatProvider provider
);
```

Public Fields

```
public static System.UInt32 MaxValue;
public static System.UInt32 MinValue;
```

UInt64 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public class UInt64
   extends System.ValueType
   implements System.IComparable, System.IFormattable, System.IConvertible
```

Public Methods

```
public final int CompareTo(
   System.Object value
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public final System.TypeCode GetTypeCode();
public System.String ToString();
public final System.String ToString(
   System.IFormatProvider provider
);
public System.String ToString(
   System.String format
);
public final System.String ToString(
   System.String format,
   System.IFormatProvider provider
);
```

Public Fields

```
public static System.UInt64 MaxValue;
public static System.UInt64 MinValue;
```

UIntPtr Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public class UIntPtr
   extends System.ValueType
   implements System.Runtime.Serialization.ISerializable
```

Public Constructors

```
public UIntPtr(
   System.UInt32 value
);
public UIntPtr(
   System.UInt64 value
);
```

Public Properties

```
public static int get_Size();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public System.String ToString();
public System.UInt32 ToUInt32();
public System.UInt64 ToUInt64();
```

Public Operators

```
public static boolean op_Equality(
   System.UIntPtr value1,
   System.UIntPtr value2
);
public static System.UInt32 op_Explicit(
   System.UIntPtr value
);
public static System.UInt64 op_Explicit(
   System.UIntPtr value
);
public static System.UIntPtr op_Explicit(
   System.UInt32 value
);
public static System.UIntPtr op_Explicit(
   System.UInt64 value
);
public static boolean op_Inequality(
   System.UIntPtr value1,
   System.UIntPtr value2
);
```

Public Fields

```
public static final System.UIntPtr Zero;
```

UnauthorizedAccessException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class UnauthorizedAccessException
   extends System.SystemException
```

Public Constructors

```
public UnauthorizedAccessException();
public UnauthorizedAccessException(
   System.String message
);
public UnauthorizedAccessException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected UnauthorizedAccessException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

UnhandledExceptionEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class UnhandledExceptionEventArgs
   extends System.EventArgs
```

Public Constructors

```
public UnhandledExceptionEventArgs(
   System.Object exception,
   boolean isTerminating
);
```

Public Properties

```
public System.Object get_ExceptionObject();
public boolean get_IsTerminating();
```

UnhandledExceptionEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void UnhandledExceptionEventHandler(
   System.Object sender,
   System.UnhandledExceptionEventArgs e
);
```

Uri Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Uri
   extends System.MarshalByRefObject
   implements System.Runtime.Serialization.ISerializable
```

Public Constructors

```
public Uri(
   System.String uriString
);
public Uri(
   System.String uriString,
   boolean dontEscape
);
public Uri(
   System.Uri baseUri,
   System.String relativeUri
);
public Uri(
   System.Uri baseUri,
   System.String relativeUri,
   boolean dontEscape
);
```

Public Properties

```
public System.String get_AbsolutePath();
public System.String get_AbsoluteUri();
public System.String get_Authority();
public System.String get_Fragment();
public System.String get_Host();
public System.UriHostNameType get_HostNameType();
public boolean get_IsDefaultPort();
public boolean get_IsFile();
public boolean get_IsLoopback();
public boolean get_IsUnc();
public System.String get_LocalPath();
public System.String get_PathAndQuery();
public int get_Port();
public System.String get_Query();
public System.String get_Scheme();
public System.String[] get_Segments();
public boolean get_UserEscaped();
public System.String get_UserInfo();
```

Public Methods

```
public static System.UriHostNameType CheckHostName(
   System.String name
);
public static boolean CheckSchemeName(
   System.String schemeName
);
```

```
public boolean Equals(
   System.Object comparand
);
public static int FromHex(
   char digit
);
public int GetHashCode();
public System.String GetLeftPart(
   System.UriPartial part
);
public static System.String HexEscape(
   char character
);
public static char HexUnescape(
   System.String pattern,
   int index
);
public static boolean IsHexDigit(
   char character
);
public static boolean IsHexEncoding(
   System.String pattern,
   int index
);
public System.String MakeRelative(
   System.Uri toUri
);
public System.String ToString();
```

Public Fields

```
public static final System.String SchemeDelimiter;
public static final System.String UriSchemeFile;
public static final System.String UriSchemeFtp;
public static final System.String UriSchemeGopher;
public static final System.String UriSchemeHttp;
public static final System.String UriSchemeHttps;
public static final System.String UriSchemeMailto;
public static final System.String UriSchemeNews;
public static final System.String UriSchemeNntp;
```

Protected Constructors

```
protected Uri(
   System.Runtime.Serialization.SerializationInfo serializationInfo,
   System.Runtime.Serialization.StreamingContext streamingContext
);
```

Protected Methods

```
protected void Canonicalize();
protected void CheckSecurity();
protected void Escape();
protected static System.String EscapeString(
   System.String str
);
protected boolean IsBadFileSystemCharacter(
   char character
);
protected static boolean IsExcludedCharacter(
   char character
);
```

```
protected boolean IsReservedCharacter(
   char character
);
protected void Parse();
protected System.String Unescape(
   System.String path
);
```

UriBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UriBuilder
   extends System.Object
```

Public Constructors

```
public UriBuilder();
public UriBuilder(
   System.String uri
);
public UriBuilder(
   System.Uri uri
);
public UriBuilder(
   System.String schemeName,
   System.String hostName
);
public UriBuilder(
   System.String scheme,
   System.String host,
   int portNumber
);
public UriBuilder(
   System.String scheme,
   System.String host,
   int port,
   System.String pathValue
);
public UriBuilder(
   System.String scheme,
   System.String host,
   int port,
   System.String path,
   System.String extraValue
);
```

Public Properties

```
public System.String get_Fragment();
public void set_Fragment(
   System.String value
);
public System.String get_Host();
public void set_Host(
   System.String value
);
public System.String get_Password();
public void set_Password(
   System.String value
);
public System.String get_Path();
public void set_Path(
   System.String value
);
public int get_Port();
public void set_Port(
```

```
   int value
);
public System.String get_Query();
public void set_Query(
   System.String value
);
public System.String get_Scheme();
public void set_Scheme(
   System.String value
);
public System.Uri get_Uri();
public System.String get_UserName();
public void set_UserName(
   System.String value
);
```

Public Methods

```
public boolean Equals(
   System.Object rparam
);
public int GetHashCode();
public System.String ToString();
```

UriFormatException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class UriFormatException
   extends System.FormatException
```

Public Constructors

```
public UriFormatException();
public UriFormatException(
   System.String textString
);
```

Protected Constructors

```
protected UriFormatException(
   System.Runtime.Serialization.SerializationInfo serializationInfo,
   System.Runtime.Serialization.StreamingContext streamingContext
);
```

UriHostNameType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum UriHostNameType
```

Members

Basic

Dns

IPv4

IPv6

Unknown

UriPartial Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum UriPartial
```

Members

Authority

Path

Scheme

ValueType Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class ValueType
   extends System.Object
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public System.String ToString();
```

Protected Constructors

```
protected ValueType();
```

Version Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Version
   extends System.Object
   implements System.ICloneable, System.IComparable
```

Public Constructors

```
public Version();
public Version(
   System.String version
);
public Version(
   int major,
   int minor
);
public Version(
   int major,
   int minor,
   int build
);
public Version(
   int major,
   int minor,
   int build,
   int revision
);
```

Public Properties

```
public int get_Build();
public int get_Major();
public int get_Minor();
public int get_Revision();
```

Public Methods

```
public final System.Object Clone();
public final int CompareTo(
   System.Object version
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public System.String ToString();
public System.String ToString(
   int fieldCount
);
```

Public Operators

```
public static boolean op_Equality(
   System.Version v1,
   System.Version v2
);
public static boolean op_GreaterThan(
   System.Version v1,
   System.Version v2
);
public static boolean op_GreaterThanOrEqual(
   System.Version v1,
   System.Version v2
);
public static boolean op_Inequality(
   System.Version v1,
   System.Version v2
);
public static boolean op_LessThan(
   System.Version v1,
   System.Version v2
);
public static boolean op_LessThanOrEqual(
   System.Version v1,
   System.Version v2
);
```

Void Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Void
   extends System.ValueType
```

WeakReference Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class WeakReference
   extends System.Object
   implements System.Runtime.Serialization.ISerializable
```

Public Constructors

```
public WeakReference(
   System.Object target
);
public WeakReference(
   System.Object target,
   boolean trackResurrection
);
```

Public Properties

```
public boolean get_IsAlive();
public System.Object get_Target();
public void set_Target(
   System.Object value
);
public boolean get_TrackResurrection();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected WeakReference(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Methods

```
protected void Finalize();
```

System.CodeDom Namespace (Visual J# Syntax)

Classes

CodeArgumentReferenceExpression

CodeArrayCreateExpression

CodeArrayIndexerExpression

CodeAssignStatement

CodeAttachEventStatement

CodeAttributeArgument

CodeAttributeArgumentCollection

CodeAttributeDeclaration

CodeAttributeDeclarationCollection

CodeBaseReferenceExpression

CodeBinaryOperatorExpression

CodeCastExpression

CodeCatchClause

CodeCatchClauseCollection

CodeComment

CodeCommentStatement

CodeCommentStatementCollection

CodeCompileUnit

CodeConditionStatement

CodeConstructor

CodeDelegateCreateExpression

CodeDelegateInvokeExpression

CodeDirectionExpression

CodeEntryPointMethod

CodeEventReferenceExpression

CodeExpression

CodeExpressionCollection

CodeExpressionStatement

CodeFieldReferenceExpression

CodeGotoStatement

CodeIndexerExpression

CodeIterationStatement

CodeLabeledStatement

CodeLinePragma

CodeMemberEvent

CodeMemberField

CodeMemberMethod

CodeMemberProperty

CodeMethodInvokeExpression

CodeMethodReferenceExpression

CodeMethodReturnStatement

CodeNamespace

CodeNamespaceCollection

CodeNamespaceImport

CodeNamespaceImportCollection

CodeObject

CodeObjectCreateExpression

CodeParameterDeclarationExpression

CodeParameterDeclarationExpressionCollection

CodePrimitiveExpression

CodePropertyReferenceExpression

CodePropertySetValueReferenceExpression

CodeRemoveEventStatement

CodeSnippetCompileUnit

CodeSnippetExpression

CodeSnippetStatement

CodeSnippetTypeMember

CodeStatement

CodeStatementCollection

CodeThisReferenceExpression

CodeThrowExceptionStatement

CodeTryCatchFinallyStatement

CodeTypeConstructor

CodeTypeDeclaration

CodeTypeDeclarationCollection

CodeTypeDelegate

CodeTypeMember

CodeTypeMemberCollection

CodeTypeOfExpression

CodeTypeReference

CodeTypeReferenceCollection

CodeTypeReferenceExpression

CodeVariableDeclarationStatement

CodeVariableReferenceExpression

Enumerations

CodeBinaryOperatorType

FieldDirection

MemberAttributes

CodeArgumentReferenceExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeArgumentReferenceExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeArgumentReferenceExpression();
public CodeArgumentReferenceExpression(
   System.String parameterName
);
```

Public Properties

```
public System.String get_ParameterName();
public void set_ParameterName(
   System.String value
);
```

CodeArrayCreateExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeArrayCreateExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeArrayCreateExpression();
public CodeArrayCreateExpression(
   System.CodeDom.CodeTypeReference createType,
   int size
);
public CodeArrayCreateExpression(
   System.CodeDom.CodeTypeReference createType,
   System.CodeDom.CodeExpression size
);
public CodeArrayCreateExpression(
   System.CodeDom.CodeTypeReference createType,
   System.CodeDom.CodeExpression[] initializers
);
public CodeArrayCreateExpression(
   System.String createType,
   int size
);
public CodeArrayCreateExpression(
   System.String createType,
   System.CodeDom.CodeExpression size
);
public CodeArrayCreateExpression(
   System.String createType,
   System.CodeDom.CodeExpression[] initializers
);
public CodeArrayCreateExpression(
   System.Type createType,
   int size
);
public CodeArrayCreateExpression(
   System.Type createType,
   System.CodeDom.CodeExpression size
);
public CodeArrayCreateExpression(
   System.Type createType,
   System.CodeDom.CodeExpression[] initializers
);
```

Public Properties

```
public System.CodeDom.CodeTypeReference get_CreateType();
public void set_CreateType(
   System.CodeDom.CodeTypeReference value
);
public System.CodeDom.CodeExpressionCollection get_Initializers();
public int get_Size();
```

```
public void set_Size(
   int value
);
public System.CodeDom.CodeExpression get_SizeExpression();
public void set_SizeExpression(
   System.CodeDom.CodeExpression value
);
```

CodeArrayIndexerExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeArrayIndexerExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeArrayIndexerExpression();
public CodeArrayIndexerExpression(
   System.CodeDom.CodeExpression targetObject,
   System.CodeDom.CodeExpression[] indices
);
```

Public Properties

```
public System.CodeDom.CodeExpressionCollection get_Indices();
public System.CodeDom.CodeExpression get_TargetObject();
public void set_TargetObject(
   System.CodeDom.CodeExpression value
);
```

CodeAssignStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeAssignStatement
   extends System.CodeDom.CodeStatement
```

Public Constructors

```
public CodeAssignStatement();
public CodeAssignStatement(
   System.CodeDom.CodeExpression left,
   System.CodeDom.CodeExpression right
);
```

Public Properties

```
public System.CodeDom.CodeExpression get_Left();
public void set_Left(
   System.CodeDom.CodeExpression value
);
public System.CodeDom.CodeExpression get_Right();
public void set_Right(
   System.CodeDom.CodeExpression value
);
```

CodeAttachEventStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeAttachEventStatement
   extends System.CodeDom.CodeStatement
```

Public Constructors

```
public CodeAttachEventStatement();
public CodeAttachEventStatement(
   System.CodeDom.CodeEventReferenceExpression eventRef,
   System.CodeDom.CodeExpression listener
);
public CodeAttachEventStatement(
   System.CodeDom.CodeExpression targetObject,
   System.String eventName,
   System.CodeDom.CodeExpression listener
);
```

Public Properties

```
public System.CodeDom.CodeEventReferenceExpression get_Event();
public void set_Event(
   System.CodeDom.CodeEventReferenceExpression value
);
public System.CodeDom.CodeExpression get_Listener();
public void set_Listener(
   System.CodeDom.CodeExpression value
);
```

CodeAttributeArgument Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeAttributeArgument
   extends System.Object
```

Public Constructors

```
public CodeAttributeArgument();
public CodeAttributeArgument(
   System.CodeDom.CodeExpression value
);
public CodeAttributeArgument(
   System.String name,
   System.CodeDom.CodeExpression value
);
```

Public Properties

```
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.CodeDom.CodeExpression get_Value();
public void set_Value(
   System.CodeDom.CodeExpression value
);
```

CodeAttributeArgumentCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CodeAttributeArgumentCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public CodeAttributeArgumentCollection();
public CodeAttributeArgumentCollection(
   System.CodeDom.CodeAttributeArgumentCollection value
);
public CodeAttributeArgumentCollection(
   System.CodeDom.CodeAttributeArgument[] value
);
```

Public Properties

```
public System.CodeDom.CodeAttributeArgument get_Item(
   int index
);
public void set_Item(
   int index,
   System.CodeDom.CodeAttributeArgument value
);
```

Public Methods

```
public int Add(
   System.CodeDom.CodeAttributeArgument value
);
public void AddRange(
   System.CodeDom.CodeAttributeArgumentCollection value
);
public void AddRange(
   System.CodeDom.CodeAttributeArgument[] value
);
public boolean Contains(
   System.CodeDom.CodeAttributeArgument value
);
public void CopyTo(
   System.CodeDom.CodeAttributeArgument[] array,
   int index
);
public int IndexOf(
   System.CodeDom.CodeAttributeArgument value
);
public void Insert(
   int index,
   System.CodeDom.CodeAttributeArgument value
);
```

```
public void Remove(
   System.CodeDom.CodeAttributeArgument value
);
```

CodeAttributeDeclaration Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeAttributeDeclaration
   extends System.Object
```

Public Constructors

```
public CodeAttributeDeclaration();
public CodeAttributeDeclaration(
   System.String name
);
public CodeAttributeDeclaration(
   System.String name,
   System.CodeDom.CodeAttributeArgument[] arguments
);
```

Public Properties

```
public System.CodeDom.CodeAttributeArgumentCollection get_Arguments();
public System.String get_Name();
public void set_Name(
   System.String value
);
```

CodeAttributeDeclarationCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeAttributeDeclarationCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public CodeAttributeDeclarationCollection();
public CodeAttributeDeclarationCollection(
   System.CodeDom.CodeAttributeDeclarationCollection value
);
public CodeAttributeDeclarationCollection(
   System.CodeDom.CodeAttributeDeclaration[] value
);
```

Public Properties

```
public System.CodeDom.CodeAttributeDeclaration get_Item(
   int index
);
public void set_Item(
   int index,
   System.CodeDom.CodeAttributeDeclaration value
);
```

Public Methods

```
public int Add(
   System.CodeDom.CodeAttributeDeclaration value
);
public void AddRange(
   System.CodeDom.CodeAttributeDeclarationCollection value
);
public void AddRange(
   System.CodeDom.CodeAttributeDeclaration[] value
);
public boolean Contains(
   System.CodeDom.CodeAttributeDeclaration value
);
public void CopyTo(
   System.CodeDom.CodeAttributeDeclaration[] array,
   int index
);
public int IndexOf(
   System.CodeDom.CodeAttributeDeclaration value
);
public void Insert(
```

```
    int index,
    System.CodeDom.CodeAttributeDeclaration value
);
public void Remove(
    System.CodeDom.CodeAttributeDeclaration value
);
```

CodeBaseReferenceExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeBaseReferenceExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeBaseReferenceExpression();
```

CodeBinaryOperatorExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeBinaryOperatorExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeBinaryOperatorExpression();
public CodeBinaryOperatorExpression(
   System.CodeDom.CodeExpression left,
   System.CodeDom.CodeBinaryOperatorType op,
   System.CodeDom.CodeExpression right
);
```

Public Properties

```
public System.CodeDom.CodeExpression get_Left();
public void set_Left(
   System.CodeDom.CodeExpression value
);
public System.CodeDom.CodeBinaryOperatorType get_Operator();
public void set_Operator(
   System.CodeDom.CodeBinaryOperatorType value
);
public System.CodeDom.CodeExpression get_Right();
public void set_Right(
   System.CodeDom.CodeExpression value
);
```

CodeBinaryOperatorType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum CodeBinaryOperatorType
```

Members

Add

Assign

BitwiseAnd

BitwiseOr

BooleanAnd

BooleanOr

Divide

GreaterThan

GreaterThanOrEqual

IdentityEquality

IdentityInequality

LessThan

LessThanOrEqual

Modulus

Multiply

Subtract

ValueEquality

CodeCastExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeCastExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeCastExpression();
public CodeCastExpression(
   System.CodeDom.CodeTypeReference targetType,
   System.CodeDom.CodeExpression expression
);
public CodeCastExpression(
   System.String targetType,
   System.CodeDom.CodeExpression expression
);
public CodeCastExpression(
   System.Type targetType,
   System.CodeDom.CodeExpression expression
);
```

Public Properties

```
public System.CodeDom.CodeExpression get_Expression();
public void set_Expression(
   System.CodeDom.CodeExpression value
);
public System.CodeDom.CodeTypeReference get_TargetType();
public void set_TargetType(
   System.CodeDom.CodeTypeReference value
);
```

CodeCatchClause Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeCatchClause
   extends System.Object
```

Public Constructors

```
public CodeCatchClause();
public CodeCatchClause(
   System.String localName
);
public CodeCatchClause(
   System.String localName,
   System.CodeDom.CodeTypeReference catchExceptionType
);
public CodeCatchClause(
   System.String localName,
   System.CodeDom.CodeTypeReference catchExceptionType,
   System.CodeDom.CodeStatement[] statements
);
```

Public Properties

```
public System.CodeDom.CodeTypeReference get_CatchExceptionType();
public void set_CatchExceptionType(
   System.CodeDom.CodeTypeReference value
);
public System.String get_LocalName();
public void set_LocalName(
   System.String value
);
public System.CodeDom.CodeStatementCollection get_Statements();
```

CodeCatchClauseCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeCatchClauseCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public CodeCatchClauseCollection();
public CodeCatchClauseCollection(
   System.CodeDom.CodeCatchClauseCollection value
);
public CodeCatchClauseCollection(
   System.CodeDom.CodeCatchClause[] value
);
```

Public Properties

```
public System.CodeDom.CodeCatchClause get_Item(
   int index
);
public void set_Item(
   int index,
   System.CodeDom.CodeCatchClause value
);
```

Public Methods

```
public int Add(
   System.CodeDom.CodeCatchClause value
);
public void AddRange(
   System.CodeDom.CodeCatchClauseCollection value
);
public void AddRange(
   System.CodeDom.CodeCatchClause[] value
);
public boolean Contains(
   System.CodeDom.CodeCatchClause value
);
public void CopyTo(
   System.CodeDom.CodeCatchClause[] array,
   int index
);
public int IndexOf(
   System.CodeDom.CodeCatchClause value
);
public void Insert(
   int index,
   System.CodeDom.CodeCatchClause value
);
```

```
public void Remove(
   System.CodeDom.CodeCatchClause value
);
```

CodeComment Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeComment
   extends System.CodeDom.CodeObject
```

Public Constructors

```
public CodeComment();
public CodeComment(
   System.String text
);
public CodeComment(
   System.String text,
   boolean docComment
);
```

Public Properties

```
public boolean get_DocComment();
public void set_DocComment(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

CodeCommentStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeCommentStatement
   extends System.CodeDom.CodeStatement
```

Public Constructors

```
public CodeCommentStatement();
public CodeCommentStatement(
   System.CodeDom.CodeComment comment
);
public CodeCommentStatement(
   System.String text
);
public CodeCommentStatement(
   System.String text,
   boolean docComment
);
```

Public Properties

```
public System.CodeDom.CodeComment get_Comment();
public void set_Comment(
   System.CodeDom.CodeComment value
);
```

CodeCommentStatementCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeCommentStatementCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public CodeCommentStatementCollection();
public CodeCommentStatementCollection(
   System.CodeDom.CodeCommentStatementCollection value
);
public CodeCommentStatementCollection(
   System.CodeDom.CodeCommentStatement[] value
);
```

Public Properties

```
public System.CodeDom.CodeCommentStatement get_Item(
   int index
);
public void set_Item(
   int index,
   System.CodeDom.CodeCommentStatement value
);
```

Public Methods

```
public int Add(
   System.CodeDom.CodeCommentStatement value
);
public void AddRange(
   System.CodeDom.CodeCommentStatementCollection value
);
public void AddRange(
   System.CodeDom.CodeCommentStatement[] value
);
public boolean Contains(
   System.CodeDom.CodeCommentStatement value
);
public void CopyTo(
   System.CodeDom.CodeCommentStatement[] array,
   int index
);
public int IndexOf(
   System.CodeDom.CodeCommentStatement value
);
public void Insert(
   int index,
   System.CodeDom.CodeCommentStatement value
);
```

```
public void Remove(
   System.CodeDom.CodeCommentStatement value
);
```

CodeCompileUnit Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeCompileUnit
   extends System.CodeDom.CodeObject
```

Public Constructors

```
public CodeCompileUnit();
```

Public Properties

```
public System.CodeDom.CodeAttributeDeclarationCollection get_AssemblyCustomAttributes();
public System.CodeDom.CodeNamespaceCollection get_Namespaces();
public System.Collections.Specialized.StringCollection get_ReferencedAssemblies();
```

CodeConditionStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeConditionStatement
   extends System.CodeDom.CodeStatement
```

Public Constructors

```
public CodeConditionStatement();
public CodeConditionStatement(
   System.CodeDom.CodeExpression condition,
   System.CodeDom.CodeStatement[] trueStatements
);
public CodeConditionStatement(
   System.CodeDom.CodeExpression condition,
   System.CodeDom.CodeStatement[] trueStatements,
   System.CodeDom.CodeStatement[] falseStatements
);
```

Public Properties

```
public System.CodeDom.CodeExpression get_Condition();
public void set_Condition(
   System.CodeDom.CodeExpression value
);
public System.CodeDom.CodeStatementCollection get_FalseStatements();
public System.CodeDom.CodeStatementCollection get_TrueStatements();
```

CodeConstructor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeConstructor
   extends System.CodeDom.CodeMemberMethod
```

Public Constructors

```
public CodeConstructor();
```

Public Properties

```
public System.CodeDom.CodeExpressionCollection get_BaseConstructorArgs();
public System.CodeDom.CodeExpressionCollection get_ChainedConstructorArgs();
```

CodeDelegateCreateExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeDelegateCreateExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeDelegateCreateExpression();
public CodeDelegateCreateExpression(
   System.CodeDom.CodeTypeReference delegateType,
   System.CodeDom.CodeExpression targetObject,
   System.String methodName
);
```

Public Properties

```
public System.CodeDom.CodeTypeReference get_DelegateType();
public void set_DelegateType(
   System.CodeDom.CodeTypeReference value
);
public System.String get_MethodName();
public void set_MethodName(
   System.String value
);
public System.CodeDom.CodeExpression get_TargetObject();
public void set_TargetObject(
   System.CodeDom.CodeExpression value
);
```

CodeDelegateInvokeExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeDelegateInvokeExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeDelegateInvokeExpression();
public CodeDelegateInvokeExpression(
   System.CodeDom.CodeExpression targetObject
);
public CodeDelegateInvokeExpression(
   System.CodeDom.CodeExpression targetObject,
   System.CodeDom.CodeExpression[] parameters
);
```

Public Properties

```
public System.CodeDom.CodeExpressionCollection get_Parameters();
public System.CodeDom.CodeExpression get_TargetObject();
public void set_TargetObject(
   System.CodeDom.CodeExpression value
);
```

CodeDirectionExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeDirectionExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeDirectionExpression();
public CodeDirectionExpression(
   System.CodeDom.FieldDirection direction,
   System.CodeDom.CodeExpression expression
);
```

Public Properties

```
public System.CodeDom.FieldDirection get_Direction();
public void set_Direction(
   System.CodeDom.FieldDirection value
);
public System.CodeDom.CodeExpression get_Expression();
public void set_Expression(
   System.CodeDom.CodeExpression value
);
```

CodeEntryPointMethod Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeEntryPointMethod
   extends System.CodeDom.CodeMemberMethod
```

Public Constructors

```
public CodeEntryPointMethod();
```

CodeEventReferenceExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeEventReferenceExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeEventReferenceExpression();
public CodeEventReferenceExpression(
   System.CodeDom.CodeExpression targetObject,
   System.String eventName
);
```

Public Properties

```
public System.String get_EventName();
public void set_EventName(
   System.String value
);
public System.CodeDom.CodeExpression get_TargetObject();
public void set_TargetObject(
   System.CodeDom.CodeExpression value
);
```

CodeExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeExpression
   extends System.CodeDom.CodeObject
```

Public Constructors

```
public CodeExpression();
```

CodeExpressionCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeExpressionCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public CodeExpressionCollection();
public CodeExpressionCollection(
   System.CodeDom.CodeExpressionCollection value
);
public CodeExpressionCollection(
   System.CodeDom.CodeExpression[] value
);
```

Public Properties

```
public System.CodeDom.CodeExpression get_Item(
   int index
);
public void set_Item(
   int index,
   System.CodeDom.CodeExpression value
);
```

Public Methods

```
public int Add(
   System.CodeDom.CodeExpression value
);
public void AddRange(
   System.CodeDom.CodeExpressionCollection value
);
public void AddRange(
   System.CodeDom.CodeExpression[] value
);
public boolean Contains(
   System.CodeDom.CodeExpression value
);
public void CopyTo(
   System.CodeDom.CodeExpression[] array,
   int index
);
public int IndexOf(
   System.CodeDom.CodeExpression value
);
public void Insert(
   int index,
   System.CodeDom.CodeExpression value
);
```

```
public void Remove(
   System.CodeDom.CodeExpression value
);
```

CodeExpressionStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeExpressionStatement
   extends System.CodeDom.CodeStatement
```

Public Constructors

```
public CodeExpressionStatement();
public CodeExpressionStatement(
   System.CodeDom.CodeExpression expression
);
```

Public Properties

```
public System.CodeDom.CodeExpression get_Expression();
public void set_Expression(
   System.CodeDom.CodeExpression value
);
```

CodeFieldReferenceExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeFieldReferenceExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeFieldReferenceExpression();
public CodeFieldReferenceExpression(
   System.CodeDom.CodeExpression targetObject,
   System.String fieldName
);
```

Public Properties

```
public System.String get_FieldName();
public void set_FieldName(
   System.String value
);
public System.CodeDom.CodeExpression get_TargetObject();
public void set_TargetObject(
   System.CodeDom.CodeExpression value
);
```

CodeGotoStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeGotoStatement
   extends System.CodeDom.CodeStatement
```

Public Constructors

```
public CodeGotoStatement(
   System.String label
);
```

Public Properties

```
public System.String get_Label();
public void set_Label(
   System.String value
);
```

CodeIndexerExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeIndexerExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeIndexerExpression();
public CodeIndexerExpression(
   System.CodeDom.CodeExpression targetObject,
   System.CodeDom.CodeExpression[] indices
);
```

Public Properties

```
public System.CodeDom.CodeExpressionCollection get_Indices();
public System.CodeDom.CodeExpression get_TargetObject();
public void set_TargetObject(
   System.CodeDom.CodeExpression value
);
```

CodeIterationStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeIterationStatement
   extends System.CodeDom.CodeStatement
```

Public Constructors

```
public CodeIterationStatement();
public CodeIterationStatement(
   System.CodeDom.CodeStatement initStatement,
   System.CodeDom.CodeExpression testExpression,
   System.CodeDom.CodeStatement incrementStatement,
   System.CodeDom.CodeStatement[] statements
);
```

Public Properties

```
public System.CodeDom.CodeStatement get_IncrementStatement();
public void set_IncrementStatement(
   System.CodeDom.CodeStatement value
);
public System.CodeDom.CodeStatement get_InitStatement();
public void set_InitStatement(
   System.CodeDom.CodeStatement value
);
public System.CodeDom.CodeStatementCollection get_Statements();
public System.CodeDom.CodeExpression get_TestExpression();
public void set_TestExpression(
   System.CodeDom.CodeExpression value
);
```

CodeLabeledStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeLabeledStatement
   extends System.CodeDom.CodeStatement
```

Public Constructors

```
public CodeLabeledStatement();
public CodeLabeledStatement(
   System.String label
);
public CodeLabeledStatement(
   System.String label,
   System.CodeDom.CodeStatement statement
);
```

Public Properties

```
public System.String get_Label();
public void set_Label(
   System.String value
);
public System.CodeDom.CodeStatement get_Statement();
public void set_Statement(
   System.CodeDom.CodeStatement value
);
```

CodeLinePragma Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeLinePragma
   extends System.Object
```

Public Constructors

```
public CodeLinePragma(
   System.String fileName,
   int lineNumber
);
```

Public Properties

```
public System.String get_FileName();
public void set_FileName(
   System.String value
);
public int get_LineNumber();
public void set_LineNumber(
   int value
);
```

CodeMemberEvent Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeMemberEvent
   extends System.CodeDom.CodeTypeMember
```

Public Constructors

```
public CodeMemberEvent();
```

Public Properties

```
public System.CodeDom.CodeTypeReferenceCollection get_ImplementationTypes();
public System.CodeDom.CodeTypeReference get_PrivateImplementationType();
public void set_PrivateImplementationType(
   System.CodeDom.CodeTypeReference value
);
public System.CodeDom.CodeTypeReference get_Type();
public void set_Type(
   System.CodeDom.CodeTypeReference value
);
```

CodeMemberField Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeMemberField
   extends System.CodeDom.CodeTypeMember
```

Public Constructors

```
public CodeMemberField();
public CodeMemberField(
   System.CodeDom.CodeTypeReference type,
   System.String name
);
public CodeMemberField(
   System.String type,
   System.String name
);
public CodeMemberField(
   System.Type type,
   System.String name
);
```

Public Properties

```
public System.CodeDom.CodeExpression get_InitExpression();
public void set_InitExpression(
   System.CodeDom.CodeExpression value
);
public System.CodeDom.CodeTypeReference get_Type();
public void set_Type(
   System.CodeDom.CodeTypeReference value
);
```

CodeMemberMethod Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeMemberMethod
   extends System.CodeDom.CodeTypeMember
```

Public Constructors

```
public CodeMemberMethod();
```

Public Properties

```
public System.CodeDom.CodeTypeReferenceCollection get_ImplementationTypes();
public System.CodeDom.CodeParameterDeclarationExpressionCollection get_Parameters();
public System.CodeDom.CodeTypeReference get_PrivateImplementationType();
public void set_PrivateImplementationType(
   System.CodeDom.CodeTypeReference value
);
public System.CodeDom.CodeTypeReference get_ReturnType();
public void set_ReturnType(
   System.CodeDom.CodeTypeReference value
);
public System.CodeDom.CodeAttributeDeclarationCollection get_ReturnTypeCustomAttributes();
public System.CodeDom.CodeStatementCollection get_Statements();
```

Public Events

```
public void add_PopulateImplementationTypes(
   System.EventHandler value
);
public void remove_PopulateImplementationTypes(
   System.EventHandler value
);
public void add_PopulateParameters(
   System.EventHandler value
);
public void remove_PopulateParameters(
   System.EventHandler value
);
public void add_PopulateStatements(
   System.EventHandler value
);
public void remove_PopulateStatements(
   System.EventHandler value
);
```

CodeMemberProperty Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeMemberProperty
   extends System.CodeDom.CodeTypeMember
```

Public Constructors

```
public CodeMemberProperty();
```

Public Properties

```
public System.CodeDom.CodeStatementCollection get_GetStatements();
public boolean get_HasGet();
public void set_HasGet(
   boolean value
);
public boolean get_HasSet();
public void set_HasSet(
   boolean value
);
public System.CodeDom.CodeTypeReferenceCollection get_ImplementationTypes();
public System.CodeDom.CodeParameterDeclarationExpressionCollection get_Parameters();
public System.CodeDom.CodeTypeReference get_PrivateImplementationType();
public void set_PrivateImplementationType(
   System.CodeDom.CodeTypeReference value
);
public System.CodeDom.CodeStatementCollection get_SetStatements();
public System.CodeDom.CodeTypeReference get_Type();
public void set_Type(
   System.CodeDom.CodeTypeReference value
);
```

CodeMethodInvokeExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeMethodInvokeExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeMethodInvokeExpression();
public CodeMethodInvokeExpression(
   System.CodeDom.CodeMethodReferenceExpression method,
   System.CodeDom.CodeExpression[] parameters
);
public CodeMethodInvokeExpression(
   System.CodeDom.CodeExpression targetObject,
   System.String methodName,
   System.CodeDom.CodeExpression[] parameters
);
```

Public Properties

```
public System.CodeDom.CodeMethodReferenceExpression get_Method();
public void set_Method(
   System.CodeDom.CodeMethodReferenceExpression value
);
public System.CodeDom.CodeExpressionCollection get_Parameters();
```

CodeMethodReferenceExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeMethodReferenceExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeMethodReferenceExpression();
public CodeMethodReferenceExpression(
   System.CodeDom.CodeExpression targetObject,
   System.String methodName
);
```

Public Properties

```
public System.String get_MethodName();
public void set_MethodName(
   System.String value
);
public System.CodeDom.CodeExpression get_TargetObject();
public void set_TargetObject(
   System.CodeDom.CodeExpression value
);
```

CodeMethodReturnStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeMethodReturnStatement
   extends System.CodeDom.CodeStatement
```

Public Constructors

```
public CodeMethodReturnStatement();
public CodeMethodReturnStatement(
   System.CodeDom.CodeExpression expression
);
```

Public Properties

```
public System.CodeDom.CodeExpression get_Expression();
public void set_Expression(
   System.CodeDom.CodeExpression value
);
```

CodeNamespace Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeNamespace
   extends System.CodeDom.CodeObject
```

Public Constructors

```
public CodeNamespace();
public CodeNamespace(
   System.String name
);
```

Public Properties

```
public System.CodeDom.CodeCommentStatementCollection get_Comments();
public System.CodeDom.CodeNamespaceImportCollection get_Imports();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.CodeDom.CodeTypeDeclarationCollection get_Types();
```

Public Events

```
public void add_PopulateComments(
   System.EventHandler value
);
public void remove_PopulateComments(
   System.EventHandler value
);
public void add_PopulateImports(
   System.EventHandler value
);
public void remove_PopulateImports(
   System.EventHandler value
);
public void add_PopulateTypes(
   System.EventHandler value
);
public void remove_PopulateTypes(
   System.EventHandler value
);
```

CodeNamespaceCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeNamespaceCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public CodeNamespaceCollection();
public CodeNamespaceCollection(
   System.CodeDom.CodeNamespaceCollection value
);
public CodeNamespaceCollection(
   System.CodeDom.CodeNamespace[] value
);
```

Public Properties

```
public System.CodeDom.CodeNamespace get_Item(
   int index
);
public void set_Item(
   int index,
   System.CodeDom.CodeNamespace value
);
```

Public Methods

```
public int Add(
   System.CodeDom.CodeNamespace value
);
public void AddRange(
   System.CodeDom.CodeNamespaceCollection value
);
public void AddRange(
   System.CodeDom.CodeNamespace[] value
);
public boolean Contains(
   System.CodeDom.CodeNamespace value
);
public void CopyTo(
   System.CodeDom.CodeNamespace[] array,
   int index
);
public int IndexOf(
   System.CodeDom.CodeNamespace value
);
public void Insert(
   int index,
   System.CodeDom.CodeNamespace value
);
```

```
public void Remove(
   System.CodeDom.CodeNamespace value
);
```

CodeNamespaceImport Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeNamespaceImport
   extends System.CodeDom.CodeObject
```

Public Constructors

```
public CodeNamespaceImport();
public CodeNamespaceImport(
   System.String nameSpace
);
```

Public Properties

```
public System.CodeDom.CodeLinePragma get_LinePragma();
public void set_LinePragma(
   System.CodeDom.CodeLinePragma value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
```

CodeNamespaceImportCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeNamespaceImportCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public CodeNamespaceImportCollection();
```

Public Properties

```
public int get_Count();
public System.CodeDom.CodeNamespaceImport get_Item(
   int index
);
public void set_Item(
   int index,
   System.CodeDom.CodeNamespaceImport value
);
```

Public Methods

```
public void Add(
   System.CodeDom.CodeNamespaceImport value
);
public void AddRange(
   System.CodeDom.CodeNamespaceImport[] value
);
public void Clear();
public System.Collections.IEnumerator GetEnumerator();
```

CodeObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeObject
   extends System.Object
```

Public Constructors

```
public CodeObject();
```

Public Properties

```
public System.Collections.IDictionary get_UserData();
```

CodeObjectCreateExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeObjectCreateExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeObjectCreateExpression();
public CodeObjectCreateExpression(
   System.CodeDom.CodeTypeReference createType,
   System.CodeDom.CodeExpression[] parameters
);
public CodeObjectCreateExpression(
   System.String createType,
   System.CodeDom.CodeExpression[] parameters
);
public CodeObjectCreateExpression(
   System.Type createType,
   System.CodeDom.CodeExpression[] parameters
);
```

Public Properties

```
public System.CodeDom.CodeTypeReference get_CreateType();
public void set_CreateType(
   System.CodeDom.CodeTypeReference value
);
public System.CodeDom.CodeExpressionCollection get_Parameters();
```

CodeParameterDeclarationExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeParameterDeclarationExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeParameterDeclarationExpression();
public CodeParameterDeclarationExpression(
   System.CodeDom.CodeTypeReference type,
   System.String name
);
public CodeParameterDeclarationExpression(
   System.String type,
   System.String name
);
public CodeParameterDeclarationExpression(
   System.Type type,
   System.String name
);
```

Public Properties

```
public System.CodeDom.CodeAttributeDeclarationCollection get_CustomAttributes();
public void set_CustomAttributes(
   System.CodeDom.CodeAttributeDeclarationCollection value
);
public System.CodeDom.FieldDirection get_Direction();
public void set_Direction(
   System.CodeDom.FieldDirection value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.CodeDom.CodeTypeReference get_Type();
public void set_Type(
   System.CodeDom.CodeTypeReference value
);
```

CodeParameterDeclarationExpressionCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeParameterDeclarationExpressionCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public CodeParameterDeclarationExpressionCollection();
public CodeParameterDeclarationExpressionCollection(
   System.CodeDom.CodeParameterDeclarationExpressionCollection value
);
public CodeParameterDeclarationExpressionCollection(
   System.CodeDom.CodeParameterDeclarationExpression[] value
);
```

Public Properties

```
public System.CodeDom.CodeParameterDeclarationExpression get_Item(
   int index
);
public void set_Item(
   int index,
   System.CodeDom.CodeParameterDeclarationExpression value
);
```

Public Methods

```
public int Add(
   System.CodeDom.CodeParameterDeclarationExpression value
);
public void AddRange(
   System.CodeDom.CodeParameterDeclarationExpressionCollection value
);
public void AddRange(
   System.CodeDom.CodeParameterDeclarationExpression[] value
);
public boolean Contains(
   System.CodeDom.CodeParameterDeclarationExpression value
);
public void CopyTo(
   System.CodeDom.CodeParameterDeclarationExpression[] array,
   int index
);
public int IndexOf(
   System.CodeDom.CodeParameterDeclarationExpression value
);
public void Insert(
```

```
    int index,
    System.CodeDom.CodeParameterDeclarationExpression value
);
public void Remove(
    System.CodeDom.CodeParameterDeclarationExpression value
);
```

CodePrimitiveExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodePrimitiveExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodePrimitiveExpression();
public CodePrimitiveExpression(
   System.Object value
);
```

Public Properties

```
public System.Object get_Value();
public void set_Value(
   System.Object value
);
```

CodePropertyReferenceExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodePropertyReferenceExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodePropertyReferenceExpression();
public CodePropertyReferenceExpression(
   System.CodeDom.CodeExpression targetObject,
   System.String propertyName
);
```

Public Properties

```
public System.String get_PropertyName();
public void set_PropertyName(
   System.String value
);
public System.CodeDom.CodeExpression get_TargetObject();
public void set_TargetObject(
   System.CodeDom.CodeExpression value
);
```

CodePropertySetValueReferenceExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodePropertySetValueReferenceExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodePropertySetValueReferenceExpression();
```

CodeRemoveEventStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeRemoveEventStatement
   extends System.CodeDom.CodeStatement
```

Public Constructors

```
public CodeRemoveEventStatement();
public CodeRemoveEventStatement(
   System.CodeDom.CodeEventReferenceExpression eventRef,
   System.CodeDom.CodeExpression listener
);
public CodeRemoveEventStatement(
   System.CodeDom.CodeExpression targetObject,
   System.String eventName,
   System.CodeDom.CodeExpression listener
);
```

Public Properties

```
public System.CodeDom.CodeEventReferenceExpression get_Event();
public void set_Event(
   System.CodeDom.CodeEventReferenceExpression value
);
public System.CodeDom.CodeExpression get_Listener();
public void set_Listener(
   System.CodeDom.CodeExpression value
);
```

CodeSnippetCompileUnit Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeSnippetCompileUnit
   extends System.CodeDom.CodeCompileUnit
```

Public Constructors

```
public CodeSnippetCompileUnit(
   System.String value
);
```

Public Properties

```
public System.CodeDom.CodeLinePragma get_LinePragma();
public void set_LinePragma(
   System.CodeDom.CodeLinePragma value
);
public System.String get_Value();
public void set_Value(
   System.String value
);
```

CodeSnippetExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeSnippetExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeSnippetExpression();
public CodeSnippetExpression(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
```

CodeSnippetStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeSnippetStatement
   extends System.CodeDom.CodeStatement
```

Public Constructors

```
public CodeSnippetStatement();
public CodeSnippetStatement(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
```

CodeSnippetTypeMember Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeSnippetTypeMember
   extends System.CodeDom.CodeTypeMember
```

Public Constructors

```
public CodeSnippetTypeMember();
public CodeSnippetTypeMember(
   System.String text
);
```

Public Properties

```
public System.String get_Text();
public void set_Text(
   System.String value
);
```

CodeStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeStatement
   extends System.CodeDom.CodeObject
```

Public Constructors

```
public CodeStatement();
```

Public Properties

```
public System.CodeDom.CodeLinePragma get_LinePragma();
public void set_LinePragma(
   System.CodeDom.CodeLinePragma value
);
```

CodeStatementCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeStatementCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public CodeStatementCollection();
public CodeStatementCollection(
   System.CodeDom.CodeStatementCollection value
);
public CodeStatementCollection(
   System.CodeDom.CodeStatement[] value
);
```

Public Properties

```
public System.CodeDom.CodeStatement get_Item(
   int index
);
public void set_Item(
   int index,
   System.CodeDom.CodeStatement value
);
```

Public Methods

```
public int Add(
   System.CodeDom.CodeExpression value
);
public int Add(
   System.CodeDom.CodeStatement value
);
public void AddRange(
   System.CodeDom.CodeStatementCollection value
);
public void AddRange(
   System.CodeDom.CodeStatement[] value
);
public boolean Contains(
   System.CodeDom.CodeStatement value
);
public void CopyTo(
   System.CodeDom.CodeStatement[] array,
   int index
);
public int IndexOf(
   System.CodeDom.CodeStatement value
);
public void Insert(
```

```
    int index,
    System.CodeDom.CodeStatement value
);
public void Remove(
    System.CodeDom.CodeStatement value
);
```

CodeThisReferenceExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeThisReferenceExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeThisReferenceExpression();
```

CodeThrowExceptionStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeThrowExceptionStatement
   extends System.CodeDom.CodeStatement
```

Public Constructors

```
public CodeThrowExceptionStatement();
public CodeThrowExceptionStatement(
   System.CodeDom.CodeExpression toThrow
);
```

Public Properties

```
public System.CodeDom.CodeExpression get_ToThrow();
public void set_ToThrow(
   System.CodeDom.CodeExpression value
);
```

CodeTryCatchFinallyStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeTryCatchFinallyStatement
   extends System.CodeDom.CodeStatement
```

Public Constructors

```
public CodeTryCatchFinallyStatement();
public CodeTryCatchFinallyStatement(
   System.CodeDom.CodeStatement[] tryStatements,
   System.CodeDom.CodeCatchClause[] catchClauses
);
public CodeTryCatchFinallyStatement(
   System.CodeDom.CodeStatement[] tryStatements,
   System.CodeDom.CodeCatchClause[] catchClauses,
   System.CodeDom.CodeStatement[] finallyStatements
);
```

Public Properties

```
public System.CodeDom.CodeCatchClauseCollection get_CatchClauses();
public System.CodeDom.CodeStatementCollection get_FinallyStatements();
public System.CodeDom.CodeStatementCollection get_TryStatements();
```

CodeTypeConstructor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeTypeConstructor
   extends System.CodeDom.CodeMemberMethod
```

Public Constructors

```
public CodeTypeConstructor();
```

CodeTypeDeclaration Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeTypeDeclaration
   extends System.CodeDom.CodeTypeMember
```

Public Constructors

```
public CodeTypeDeclaration();
public CodeTypeDeclaration(
   System.String name
);
```

Public Properties

```
public System.CodeDom.CodeTypeReferenceCollection get_BaseTypes();
public boolean get_IsClass();
public void set_IsClass(
   boolean value
);
public boolean get_IsEnum();
public void set_IsEnum(
   boolean value
);
public boolean get_IsInterface();
public void set_IsInterface(
   boolean value
);
public boolean get_IsStruct();
public void set_IsStruct(
   boolean value
);
public System.CodeDom.CodeTypeMemberCollection get_Members();
public System.Reflection.TypeAttributes get_TypeAttributes();
public void set_TypeAttributes(
   System.Reflection.TypeAttributes value
);
```

Public Events

```
public void add_PopulateBaseTypes(
   System.EventHandler value
);
public void remove_PopulateBaseTypes(
   System.EventHandler value
);
public void add_PopulateMembers(
   System.EventHandler value
);
public void remove_PopulateMembers(
   System.EventHandler value
);
```

CodeTypeDeclarationCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeTypeDeclarationCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public CodeTypeDeclarationCollection();
public CodeTypeDeclarationCollection(
   System.CodeDom.CodeTypeDeclarationCollection value
);
public CodeTypeDeclarationCollection(
   System.CodeDom.CodeTypeDeclaration[] value
);
```

Public Properties

```
public System.CodeDom.CodeTypeDeclaration get_Item(
   int index
);
public void set_Item(
   int index,
   System.CodeDom.CodeTypeDeclaration value
);
```

Public Methods

```
public int Add(
   System.CodeDom.CodeTypeDeclaration value
);
public void AddRange(
   System.CodeDom.CodeTypeDeclarationCollection value
);
public void AddRange(
   System.CodeDom.CodeTypeDeclaration[] value
);
public boolean Contains(
   System.CodeDom.CodeTypeDeclaration value
);
public void CopyTo(
   System.CodeDom.CodeTypeDeclaration[] array,
   int index
);
public int IndexOf(
   System.CodeDom.CodeTypeDeclaration value
);
public void Insert(
   int index,
   System.CodeDom.CodeTypeDeclaration value
);
```

```
public void Remove(
   System.CodeDom.CodeTypeDeclaration value
);
```

CodeTypeDelegate Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeTypeDelegate
   extends System.CodeDom.CodeTypeDeclaration
```

Public Constructors

```
public CodeTypeDelegate();
public CodeTypeDelegate(
   System.String name
);
```

Public Properties

```
public System.CodeDom.CodeParameterDeclarationExpressionCollection get_Parameters();
public System.CodeDom.CodeTypeReference get_ReturnType();
public void set_ReturnType(
   System.CodeDom.CodeTypeReference value
);
```

CodeTypeMember Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeTypeMember
   extends System.CodeDom.CodeObject
```

Public Constructors

```
public CodeTypeMember();
```

Public Properties

```
public System.CodeDom.MemberAttributes get_Attributes();
public void set_Attributes(
   System.CodeDom.MemberAttributes value
);
public System.CodeDom.CodeCommentStatementCollection get_Comments();
public System.CodeDom.CodeAttributeDeclarationCollection get_CustomAttributes();
public void set_CustomAttributes(
   System.CodeDom.CodeAttributeDeclarationCollection value
);
public System.CodeDom.CodeLinePragma get_LinePragma();
public void set_LinePragma(
   System.CodeDom.CodeLinePragma value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
```

CodeTypeMemberCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CodeTypeMemberCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public CodeTypeMemberCollection();
public CodeTypeMemberCollection(
   System.CodeDom.CodeTypeMemberCollection value
);
public CodeTypeMemberCollection(
   System.CodeDom.CodeTypeMember[] value
);
```

Public Properties

```
public System.CodeDom.CodeTypeMember get_Item(
   int index
);
public void set_Item(
   int index,
   System.CodeDom.CodeTypeMember value
);
```

Public Methods

```
public int Add(
   System.CodeDom.CodeTypeMember value
);
public void AddRange(
   System.CodeDom.CodeTypeMemberCollection value
);
public void AddRange(
   System.CodeDom.CodeTypeMember[] value
);
public boolean Contains(
   System.CodeDom.CodeTypeMember value
);
public void CopyTo(
   System.CodeDom.CodeTypeMember[] array,
   int index
);
public int IndexOf(
   System.CodeDom.CodeTypeMember value
);
public void Insert(
   int index,
   System.CodeDom.CodeTypeMember value
);
```

```
public void Remove(
   System.CodeDom.CodeTypeMember value
);
```

CodeTypeOfExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeTypeOfExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeTypeOfExpression();
public CodeTypeOfExpression(
   System.CodeDom.CodeTypeReference type
);
public CodeTypeOfExpression(
   System.String type
);
public CodeTypeOfExpression(
   System.Type type
);
```

Public Properties

```
public System.CodeDom.CodeTypeReference get_Type();
public void set_Type(
   System.CodeDom.CodeTypeReference value
);
```

CodeTypeReference Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeTypeReference
   extends System.CodeDom.CodeObject
```

Public Constructors

```
public CodeTypeReference(
   System.String typeName
);
public CodeTypeReference(
   System.Type type
);
public CodeTypeReference(
   System.CodeDom.CodeTypeReference arrayType,
   int rank
);
public CodeTypeReference(
   System.String baseType,
   int rank
);
```

Public Properties

```
public System.CodeDom.CodeTypeReference get_ArrayElementType();
public void set_ArrayElementType(
   System.CodeDom.CodeTypeReference value
);
public int get_ArrayRank();
public void set_ArrayRank(
   int value
);
public System.String get_BaseType();
public void set_BaseType(
   System.String value
);
```

CodeTypeReferenceCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CodeTypeReferenceCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public CodeTypeReferenceCollection();
public CodeTypeReferenceCollection(
   System.CodeDom.CodeTypeReferenceCollection value
);
public CodeTypeReferenceCollection(
   System.CodeDom.CodeTypeReference[] value
);
```

Public Properties

```
public System.CodeDom.CodeTypeReference get_Item(
   int index
);
public void set_Item(
   int index,
   System.CodeDom.CodeTypeReference value
);
```

Public Methods

```
public int Add(
   System.CodeDom.CodeTypeReference value
);
public void Add(
   System.String value
);
public void Add(
   System.Type value
);
public void AddRange(
   System.CodeDom.CodeTypeReferenceCollection value
);
public void AddRange(
   System.CodeDom.CodeTypeReference[] value
);
public boolean Contains(
   System.CodeDom.CodeTypeReference value
);
public void CopyTo(
   System.CodeDom.CodeTypeReference[] array,
   int index
);
public int IndexOf(
```

```
   System.CodeDom.CodeTypeReference value
);
public void Insert(
   int index,
   System.CodeDom.CodeTypeReference value
);
public void Remove(
   System.CodeDom.CodeTypeReference value
);
```

CodeTypeReferenceExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeTypeReferenceExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeTypeReferenceExpression();
public CodeTypeReferenceExpression(
   System.CodeDom.CodeTypeReference type
);
public CodeTypeReferenceExpression(
   System.String type
);
public CodeTypeReferenceExpression(
   System.Type type
);
```

Public Properties

```
public System.CodeDom.CodeTypeReference get_Type();
public void set_Type(
   System.CodeDom.CodeTypeReference value
);
```

CodeVariableDeclarationStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
public class CodeVariableDeclarationStatement
   extends System.CodeDom.CodeStatement
```

Public Constructors

```
public CodeVariableDeclarationStatement();
public CodeVariableDeclarationStatement(
   System.CodeDom.CodeTypeReference type,
   System.String name
);
public CodeVariableDeclarationStatement(
   System.String type,
   System.String name
);
public CodeVariableDeclarationStatement(
   System.Type type,
   System.String name
);
public CodeVariableDeclarationStatement(
   System.CodeDom.CodeTypeReference type,
   System.String name,
   System.CodeDom.CodeExpression initExpression
);
public CodeVariableDeclarationStatement(
   System.String type,
   System.String name,
   System.CodeDom.CodeExpression initExpression
);
public CodeVariableDeclarationStatement(
   System.Type type,
   System.String name,
   System.CodeDom.CodeExpression initExpression
);
```

Public Properties

```
public System.CodeDom.CodeExpression get_InitExpression();
public void set_InitExpression(
   System.CodeDom.CodeExpression value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.CodeDom.CodeTypeReference get_Type();
public void set_Type(
   System.CodeDom.CodeTypeReference value
);
```

CodeVariableReferenceExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDispatch) *
/
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CodeVariableReferenceExpression
   extends System.CodeDom.CodeExpression
```

Public Constructors

```
public CodeVariableReferenceExpression();
public CodeVariableReferenceExpression(
   System.String variableName
);
```

Public Properties

```
public System.String get_VariableName();
public void set_VariableName(
   System.String value
);
```

FieldDirection Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum FieldDirection
```

Members

In

Out

Ref

MemberAttributes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum MemberAttributes
```

Members

Abstract

AccessMask

Assembly

Const

Family

FamilyAndAssembly

FamilyOrAssembly

Final

New

Overloaded

Override

Private

Public

ScopeMask

Static

VTableMask

System.CodeDom.Compiler Namespace (Visual J# Syntax)

Classes

CodeCompiler

CodeDomProvider

CodeGenerator

CodeGeneratorOptions

CodeParser

CompilerError

CompilerErrorCollection

CompilerParameters

CompilerResults

Executor

IndentedTextWriter

TempFileCollection

Interfaces

ICodeCompiler

ICodeGenerator

ICodeParser

Enumerations

GeneratorSupport

LanguageOptions

CodeCompiler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class CodeCompiler
   extends System.CodeDom.Compiler.CodeGenerator
   implements System.CodeDom.Compiler.ICodeCompiler
```

Protected Constructors

```
protected CodeCompiler();
```

Protected Properties

```
protected abstract System.String get_CompilerName();
protected abstract System.String get_FileExtension();
```

Protected Methods

```
protected abstract System.String CmdArgsFromParameters(
   System.CodeDom.Compiler.CompilerParameters options
);
protected System.CodeDom.Compiler.CompilerResults FromDom(
   System.CodeDom.Compiler.CompilerParameters options,
   System.CodeDom.CodeCompileUnit e
);
protected System.CodeDom.Compiler.CompilerResults FromDomBatch(
   System.CodeDom.Compiler.CompilerParameters options,
   System.CodeDom.CodeCompileUnit[] ea
);
protected System.CodeDom.Compiler.CompilerResults FromFile(
   System.CodeDom.Compiler.CompilerParameters options,
   System.String fileName
);
protected System.CodeDom.Compiler.CompilerResults FromFileBatch(
   System.CodeDom.Compiler.CompilerParameters options,
   System.String[] fileNames
);
protected System.CodeDom.Compiler.CompilerResults FromSource(
   System.CodeDom.Compiler.CompilerParameters options,
   System.String source
);
protected System.CodeDom.Compiler.CompilerResults FromSourceBatch(
   System.CodeDom.Compiler.CompilerParameters options,
   System.String[] sources
);
protected System.String GetResponseFileCmdArgs(
   System.CodeDom.Compiler.CompilerParameters options,
   System.String cmdArgs
);
protected static System.String JoinStringArray(
   System.String[] sa,
   System.String separator
);
protected abstract void ProcessCompilerOutputLine(
   System.CodeDom.Compiler.CompilerResults results,
   System.String line
```

```
);
```

CodeDomProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public abstract class CodeDomProvider
   extends System.ComponentModel.Component
```

Public Properties

```
public System.String get_FileExtension();
public System.CodeDom.Compiler.LanguageOptions get_LanguageOptions();
```

Public Methods

```
public abstract System.CodeDom.Compiler.ICodeCompiler CreateCompiler();
public abstract System.CodeDom.Compiler.ICodeGenerator CreateGenerator();
public System.CodeDom.Compiler.ICodeGenerator CreateGenerator(
   System.IO.TextWriter output
);
public System.CodeDom.Compiler.ICodeGenerator CreateGenerator(
   System.String fileName
);
public System.CodeDom.Compiler.ICodeParser CreateParser();
public System.ComponentModel.TypeConverter GetConverter(
   System.Type type
);
```

Protected Constructors

```
protected CodeDomProvider();
```

CodeGenerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class CodeGenerator
   extends System.Object
   implements System.CodeDom.Compiler.ICodeGenerator
```

Public Methods

```
public static boolean IsValidLanguageIndependentIdentifier(
   System.String value
);
```

Protected Constructors

```
protected CodeGenerator();
```

Protected Properties

```
protected System.CodeDom.CodeTypeMember get_CurrentMember();
protected System.String get_CurrentMemberName();
protected System.String get_CurrentTypeName();
protected int get_Indent();
protected void set_Indent(
   int value
);
protected boolean get_IsCurrentClass();
protected boolean get_IsCurrentDelegate();
protected boolean get_IsCurrentEnum();
protected boolean get_IsCurrentInterface();
protected boolean get_IsCurrentStruct();
protected abstract System.String get_NullToken();
protected System.CodeDom.Compiler.CodeGeneratorOptions get_Options();
protected System.IO.TextWriter get_Output();
```

Protected Methods

```
protected void ContinueOnNewLine(
   System.String st
);
protected abstract System.String CreateEscapedIdentifier(
   System.String value
);
protected abstract System.String CreateValidIdentifier(
   System.String value
);
protected abstract void GenerateArgumentReferenceExpression(
   System.CodeDom.CodeArgumentReferenceExpression e
);
protected abstract void GenerateArrayCreateExpression(
   System.CodeDom.CodeArrayCreateExpression e
);
protected abstract void GenerateArrayIndexerExpression(
   System.CodeDom.CodeArrayIndexerExpression e
```

```
);
protected abstract void GenerateAssignStatement(
   System.CodeDom.CodeAssignStatement e
);
protected abstract void GenerateAttachEventStatement(
   System.CodeDom.CodeAttachEventStatement e
);
protected abstract void GenerateAttributeDeclarationsEnd(
   System.CodeDom.CodeAttributeDeclarationCollection attributes
);
protected abstract void GenerateAttributeDeclarationsStart(
   System.CodeDom.CodeAttributeDeclarationCollection attributes
);
protected abstract void GenerateBaseReferenceExpression(
   System.CodeDom.CodeBaseReferenceExpression e
);
protected void GenerateBinaryOperatorExpression(
   System.CodeDom.CodeBinaryOperatorExpression e
);
protected abstract void GenerateCastExpression(
   System.CodeDom.CodeCastExpression e
);
protected abstract void GenerateComment(
   System.CodeDom.CodeComment e
);
protected void GenerateCommentStatement(
   System.CodeDom.CodeCommentStatement e
);
protected void GenerateCommentStatements(
   System.CodeDom.CodeCommentStatementCollection e
);
protected void GenerateCompileUnit(
   System.CodeDom.CodeCompileUnit e
);
protected void GenerateCompileUnitEnd(
   System.CodeDom.CodeCompileUnit e
);
protected void GenerateCompileUnitStart(
   System.CodeDom.CodeCompileUnit e
);
protected abstract void GenerateConditionStatement(
   System.CodeDom.CodeConditionStatement e
);
protected abstract void GenerateConstructor(
   System.CodeDom.CodeConstructor e,
   System.CodeDom.CodeTypeDeclaration c
);
protected void GenerateDecimalValue(
   System.Decimal d
);
protected abstract void GenerateDelegateCreateExpression(
   System.CodeDom.CodeDelegateCreateExpression e
);
protected abstract void GenerateDelegateInvokeExpression(
   System.CodeDom.CodeDelegateInvokeExpression e
);
protected void GenerateDirectionExpression(
   System.CodeDom.CodeDirectionExpression e
);
protected void GenerateDoubleValue(
   double d
);
protected abstract void GenerateEntryPointMethod(
   System.CodeDom.CodeEntryPointMethod e,
   System.CodeDom.CodeTypeDeclaration c
);
protected abstract void GenerateEvent(
   System.CodeDom.CodeMemberEvent e,
```

```
    System.CodeDom.CodeTypeDeclaration c
  );
  protected abstract void GenerateEventReferenceExpression(
    System.CodeDom.CodeEventReferenceExpression e
  );
  protected void GenerateExpression(
    System.CodeDom.CodeExpression e
  );
  protected abstract void GenerateExpressionStatement(
    System.CodeDom.CodeExpressionStatement e
  );
  protected abstract void GenerateField(
    System.CodeDom.CodeMemberField e
  );
  protected abstract void GenerateFieldReferenceExpression(
    System.CodeDom.CodeFieldReferenceExpression e
  );
  protected abstract void GenerateGotoStatement(
    System.CodeDom.CodeGotoStatement e
  );
  protected abstract void GenerateIndexerExpression(
    System.CodeDom.CodeIndexerExpression e
  );
  protected abstract void GenerateIterationStatement(
    System.CodeDom.CodeIterationStatement e
  );
  protected abstract void GenerateLabeledStatement(
    System.CodeDom.CodeLabeledStatement e
  );
  protected abstract void GenerateLinePragmaEnd(
    System.CodeDom.CodeLinePragma e
  );
  protected abstract void GenerateLinePragmaStart(
    System.CodeDom.CodeLinePragma e
  );
  protected abstract void GenerateMethod(
    System.CodeDom.CodeMemberMethod e,
    System.CodeDom.CodeTypeDeclaration c
  );
  protected abstract void GenerateMethodInvokeExpression(
    System.CodeDom.CodeMethodInvokeExpression e
  );
  protected abstract void GenerateMethodReferenceExpression(
    System.CodeDom.CodeMethodReferenceExpression e
  );
  protected abstract void GenerateMethodReturnStatement(
    System.CodeDom.CodeMethodReturnStatement e
  );
  protected void GenerateNamespace(
    System.CodeDom.CodeNamespace e
  );
  protected abstract void GenerateNamespaceEnd(
    System.CodeDom.CodeNamespace e
  );
  protected abstract void GenerateNamespaceImport(
    System.CodeDom.CodeNamespaceImport e
  );
  protected void GenerateNamespaceImports(
    System.CodeDom.CodeNamespace e
  );
  protected void GenerateNamespaces(
    System.CodeDom.CodeCompileUnit e
  );
  protected abstract void GenerateNamespaceStart(
    System.CodeDom.CodeNamespace e
  );
  protected abstract void GenerateObjectCreateExpression(
    System.CodeDom.CodeObjectCreateExpression e
```

```
);
protected void GenerateParameterDeclarationExpression(
   System.CodeDom.CodeParameterDeclarationExpression e
);
protected void GeneratePrimitiveExpression(
   System.CodeDom.CodePrimitiveExpression e
);
protected abstract void GenerateProperty(
   System.CodeDom.CodeMemberProperty e,
   System.CodeDom.CodeTypeDeclaration c
);
protected abstract void GeneratePropertyReferenceExpression(
   System.CodeDom.CodePropertyReferenceExpression e
);
protected abstract void GeneratePropertySetValueReferenceExpression(
   System.CodeDom.CodePropertySetValueReferenceExpression e
);
protected abstract void GenerateRemoveEventStatement(
   System.CodeDom.CodeRemoveEventStatement e
);
protected void GenerateSingleFloatValue(
   float s
);
protected void GenerateSnippetCompileUnit(
   System.CodeDom.CodeSnippetCompileUnit e
);
protected abstract void GenerateSnippetExpression(
   System.CodeDom.CodeSnippetExpression e
);
protected abstract void GenerateSnippetMember(
   System.CodeDom.CodeSnippetTypeMember e
);
protected void GenerateSnippetStatement(
   System.CodeDom.CodeSnippetStatement e
);
protected void GenerateStatement(
   System.CodeDom.CodeStatement e
);
protected void GenerateStatements(
   System.CodeDom.CodeStatementCollection stms
);
protected abstract void GenerateThisReferenceExpression(
   System.CodeDom.CodeThisReferenceExpression e
);
protected abstract void GenerateThrowExceptionStatement(
   System.CodeDom.CodeThrowExceptionStatement e
);
protected abstract void GenerateTryCatchFinallyStatement(
   System.CodeDom.CodeTryCatchFinallyStatement e
);
protected abstract void GenerateTypeConstructor(
   System.CodeDom.CodeTypeConstructor e
);
protected abstract void GenerateTypeEnd(
   System.CodeDom.CodeTypeDeclaration e
);
protected void GenerateTypeOfExpression(
   System.CodeDom.CodeTypeOfExpression e
);
protected void GenerateTypeReferenceExpression(
   System.CodeDom.CodeTypeReferenceExpression e
);
protected void GenerateTypes(
   System.CodeDom.CodeNamespace e
);
protected abstract void GenerateTypeStart(
   System.CodeDom.CodeTypeDeclaration e
);
```

```
protected abstract void GenerateVariableDeclarationStatement(
   System.CodeDom.CodeVariableDeclarationStatement e
);
protected abstract void GenerateVariableReferenceExpression(
   System.CodeDom.CodeVariableReferenceExpression e
);
protected abstract System.String GetTypeOutput(
   System.CodeDom.CodeTypeReference value
);
protected abstract boolean IsValidIdentifier(
   System.String value
);
protected void OutputAttributeArgument(
   System.CodeDom.CodeAttributeArgument arg
);
protected void OutputAttributeDeclarations(
   System.CodeDom.CodeAttributeDeclarationCollection attributes
);
protected void OutputDirection(
   System.CodeDom.FieldDirection dir
);
protected void OutputExpressionList(
   System.CodeDom.CodeExpressionCollection expressions
);
protected void OutputExpressionList(
   System.CodeDom.CodeExpressionCollection expressions,
   boolean newlineBetweenItems
);
protected void OutputFieldScopeModifier(
   System.CodeDom.MemberAttributes attributes
);
protected void OutputIdentifier(
   System.String ident
);
protected void OutputMemberAccessModifier(
   System.CodeDom.MemberAttributes attributes
);
protected void OutputMemberScopeModifier(
   System.CodeDom.MemberAttributes attributes
);
protected void OutputOperator(
   System.CodeDom.CodeBinaryOperatorType op
);
protected void OutputParameters(
   System.CodeDom.CodeParameterDeclarationExpressionCollection parameters
);
protected abstract void OutputType(
   System.CodeDom.CodeTypeReference typeRef
);
protected void OutputTypeAttributes(
   System.Reflection.TypeAttributes attributes,
   boolean isStruct,
   boolean isEnum
);
protected void OutputTypeNamePair(
   System.CodeDom.CodeTypeReference typeRef,
   System.String name
);
protected abstract System.String QuoteSnippetString(
   System.String value
);
protected abstract boolean Supports(
   System.CodeDom.Compiler.GeneratorSupport support
);
protected void ValidateIdentifier(
   System.String value
);
```

CodeGeneratorOptions Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CodeGeneratorOptions
   extends System.Object
```

Public Constructors

```
public CodeGeneratorOptions();
```

Public Properties

```
public boolean get_BlankLinesBetweenMembers();
public void set_BlankLinesBetweenMembers(
   boolean value
);
public System.String get_BracingStyle();
public void set_BracingStyle(
   System.String value
);
public boolean get_ElseOnClosing();
public void set_ElseOnClosing(
   boolean value
);
public System.String get_IndentString();
public void set_IndentString(
   System.String value
);
public System.Object get_Item(
   System.String index
);
public void set_Item(
   System.String index,
   System.Object value
);
```

CodeParser Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class CodeParser
   extends System.Object
   implements System.CodeDom.Compiler.ICodeParser
```

Public Methods

```
public abstract System.CodeDom.CodeCompileUnit Parse(
   System.IO.TextReader codeStream
);
```

Protected Constructors

```
protected CodeParser();
```

CompilerError Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CompilerError
   extends System.Object
```

Public Constructors

```
public CompilerError();
public CompilerError(
   System.String fileName,
   int line,
   int column,
   System.String errorNumber,
   System.String errorText
);
```

Public Properties

```
public int get_Column();
public void set_Column(
   int value
);
public System.String get_ErrorNumber();
public void set_ErrorNumber(
   System.String value
);
public System.String get_ErrorText();
public void set_ErrorText(
   System.String value
);
public System.String get_FileName();
public void set_FileName(
   System.String value
);
public boolean get_IsWarning();
public void set_IsWarning(
   boolean value
);
public int get_Line();
public void set_Line(
   int value
);
```

Public Methods

```
public System.String ToString();
```

CompilerErrorCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CompilerErrorCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public CompilerErrorCollection();
public CompilerErrorCollection(
   System.CodeDom.Compiler.CompilerErrorCollection value
);
public CompilerErrorCollection(
   System.CodeDom.Compiler.CompilerError[] value
);
```

Public Properties

```
public boolean get_HasErrors();
public boolean get_HasWarnings();
public System.CodeDom.Compiler.CompilerError get_Item(
   int index
);
public void set_Item(
   int index,
   System.CodeDom.Compiler.CompilerError value
);
```

Public Methods

```
public int Add(
   System.CodeDom.Compiler.CompilerError value
);
public void AddRange(
   System.CodeDom.Compiler.CompilerErrorCollection value
);
public void AddRange(
   System.CodeDom.Compiler.CompilerError[] value
);
public boolean Contains(
   System.CodeDom.Compiler.CompilerError value
);
public void CopyTo(
   System.CodeDom.Compiler.CompilerError[] array,
   int index
);
public int IndexOf(
   System.CodeDom.Compiler.CompilerError value
);
public void Insert(
   int index,
   System.CodeDom.Compiler.CompilerError value
);
public void Remove(
```

```
    System.CodeDom.Compiler.CompilerError value
);
```

CompilerParameters Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class CompilerParameters
   extends System.Object
```

Public Constructors

```
public CompilerParameters();
public CompilerParameters(
   System.String[] assemblyNames
);
public CompilerParameters(
   System.String[] assemblyNames,
   System.String outputName
);
public CompilerParameters(
   System.String[] assemblyNames,
   System.String outputName,
   boolean includeDebugInformation
);
```

Public Properties

```
public System.String get_CompilerOptions();
public void set_CompilerOptions(
   System.String value
);
public boolean get_GenerateExecutable();
public void set_GenerateExecutable(
   boolean value
);
public boolean get_GenerateInMemory();
public void set_GenerateInMemory(
   boolean value
);
public boolean get_IncludeDebugInformation();
public void set_IncludeDebugInformation(
   boolean value
);
public System.String get_MainClass();
public void set_MainClass(
   System.String value
);
public System.String get_OutputAssembly();
public void set_OutputAssembly(
   System.String value
);
public System.Collections.Specialized.StringCollection get_ReferencedAssemblies();
public System.CodeDom.Compiler.TempFileCollection get_TempFiles();
public void set_TempFiles(
   System.CodeDom.Compiler.TempFileCollection value
);
public boolean get_TreatWarningsAsErrors();
public void set_TreatWarningsAsErrors(
   boolean value
);
```

```
public System.IntPtr get_UserToken();
public void set_UserToken(
   System.IntPtr value
);
public int get_WarningLevel();
public void set_WarningLevel(
   int value
);
public System.String get_Win32Resource();
public void set_Win32Resource(
   System.String value
);
```

CompilerResults Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CompilerResults
   extends System.Object
```

Public Constructors

```
public CompilerResults(
   System.CodeDom.Compiler.TempFileCollection tempFiles
);
```

Public Properties

```
public System.Reflection.Assembly get_CompiledAssembly();
public void set_CompiledAssembly(
   System.Reflection.Assembly value
);
public System.CodeDom.Compiler.CompilerErrorCollection get_Errors();
public int get_NativeCompilerReturnValue();
public void set_NativeCompilerReturnValue(
   int value
);
public System.Collections.Specialized.StringCollection get_Output();
public System.String get_PathToAssembly();
public void set_PathToAssembly(
   System.String value
);
public System.CodeDom.Compiler.TempFileCollection get_TempFiles();
public void set_TempFiles(
   System.CodeDom.Compiler.TempFileCollection value
);
```

Executor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Executor
   extends System.Object
```

Public Methods

```
public static void ExecWait(
   System.String cmd,
   System.CodeDom.Compiler.TempFileCollection tempFiles
);
public static int ExecWaitWithCapture(
   System.String cmd,
   System.CodeDom.Compiler.TempFileCollection tempFiles,
   System.String& outputName,
   System.String& errorName
);
public static int ExecWaitWithCapture(
   System.IntPtr userToken,
   System.String cmd,
   System.CodeDom.Compiler.TempFileCollection tempFiles,
   System.String& outputName,
   System.String& errorName
);
public static int ExecWaitWithCapture(
   System.String cmd,
   System.String currentDir,
   System.CodeDom.Compiler.TempFileCollection tempFiles,
   System.String& outputName,
   System.String& errorName
);
public static int ExecWaitWithCapture(
   System.IntPtr userToken,
   System.String cmd,
   System.String currentDir,
   System.CodeDom.Compiler.TempFileCollection tempFiles,
   System.String& outputName,
   System.String& errorName
);
```

GeneratorSupport Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum GeneratorSupport
```

Members

ArraysOfArrays

AssemblyAttributes

ChainedConstructorArguments

ComplexExpressions

DeclareDelegates

DeclareEnums

DeclareEvents

DeclareInterfaces

DeclareValueTypes

EntryPointMethod

GotoStatements

MultidimensionalArrays

MultipleInterfaceMembers

NestedTypes

ParameterAttributes

PublicStaticMembers

ReferenceParameters

ReturnTypeAttributes

StaticConstructors

TryCatchStatements

Win32Resources

ICodeCompiler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICodeCompiler
```

Public Methods

```
System.CodeDom.Compiler.CompilerResults CompileAssemblyFromDom(
   System.CodeDom.Compiler.CompilerParameters options,
   System.CodeDom.CodeCompileUnit compilationUnit
);
System.CodeDom.Compiler.CompilerResults CompileAssemblyFromDomBatch(
   System.CodeDom.Compiler.CompilerParameters options,
   System.CodeDom.CodeCompileUnit[] compilationUnits
);
System.CodeDom.Compiler.CompilerResults CompileAssemblyFromFile(
   System.CodeDom.Compiler.CompilerParameters options,
   System.String fileName
);
System.CodeDom.Compiler.CompilerResults CompileAssemblyFromFileBatch(
   System.CodeDom.Compiler.CompilerParameters options,
   System.String[] fileNames
);
System.CodeDom.Compiler.CompilerResults CompileAssemblyFromSource(
   System.CodeDom.Compiler.CompilerParameters options,
   System.String source
);
System.CodeDom.Compiler.CompilerResults CompileAssemblyFromSourceBatch(
   System.CodeDom.Compiler.CompilerParameters options,
   System.String[] sources
);
```

ICodeGenerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICodeGenerator
```

Public Methods

```
System.String CreateEscapedIdentifier(
   System.String value
);
System.String CreateValidIdentifier(
   System.String value
);
void GenerateCodeFromCompileUnit(
   System.CodeDom.CodeCompileUnit e,
   System.IO.TextWriter w,
   System.CodeDom.Compiler.CodeGeneratorOptions o
);
void GenerateCodeFromExpression(
   System.CodeDom.CodeExpression e,
   System.IO.TextWriter w,
   System.CodeDom.Compiler.CodeGeneratorOptions o
);
void GenerateCodeFromNamespace(
   System.CodeDom.CodeNamespace e,
   System.IO.TextWriter w,
   System.CodeDom.Compiler.CodeGeneratorOptions o
);
void GenerateCodeFromStatement(
   System.CodeDom.CodeStatement e,
   System.IO.TextWriter w,
   System.CodeDom.Compiler.CodeGeneratorOptions o
);
void GenerateCodeFromType(
   System.CodeDom.CodeTypeDeclaration e,
   System.IO.TextWriter w,
   System.CodeDom.Compiler.CodeGeneratorOptions o
);
System.String GetTypeOutput(
   System.CodeDom.CodeTypeReference type
);
boolean IsValidIdentifier(
   System.String value
);
boolean Supports(
   System.CodeDom.Compiler.GeneratorSupport supports
);
void ValidateIdentifier(
   System.String value
);
```

ICodeParser Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICodeParser
```

Public Methods

```
System.CodeDom.CodeCompileUnit Parse(
   System.IO.TextReader codeStream
);
```

IndentedTextWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class IndentedTextWriter
   extends System.IO.TextWriter
```

Public Constructors

```
public IndentedTextWriter(
   System.IO.TextWriter writer
);
public IndentedTextWriter(
   System.IO.TextWriter writer,
   System.String tabString
);
```

Public Properties

```
public System.Text.Encoding get_Encoding();
public int get_Indent();
public void set_Indent(
   int value
);
public System.IO.TextWriter get_InnerWriter();
public System.String get_NewLine();
public void set_NewLine(
   System.String value
);
```

Public Methods

```
public void Close();
public void Flush();
public void Write(
   boolean value
);
public void Write(
   char value
);
public void Write(
   char[] buffer
);
public void Write(
   double value
);
public void Write(
   int value
);
public void Write(
   long value
);
public void Write(
   float value
);
public void Write(
   System.Object value
```

```
);
public void Write(
   System.String s
);
public void Write(
   System.String format,
   System.Object arg0
);
public void Write(
   System.String format,
   System.Object[] arg
);
public void Write(
   char[] buffer,
   int index,
   int count
);
public void Write(
   System.String format,
   System.Object arg0,
   System.Object arg1
);
public void WriteLine();
public void WriteLine(
   boolean value
);
public void WriteLine(
   char value
);
public void WriteLine(
   char[] buffer
);
public void WriteLine(
   double value
);
public void WriteLine(
   int value
);
public void WriteLine(
   long value
);
public void WriteLine(
   float value
);
public void WriteLine(
   System.Object value
);
public void WriteLine(
   System.String s
);
public void WriteLine(
   System.String format,
   System.Object arg0
);
public void WriteLine(
   System.String format,
   System.Object[] arg
);
public void WriteLine(
   char[] buffer,
   int index,
   int count
);
public void WriteLine(
   System.String format,
   System.Object arg0,
   System.Object arg1
);
```

```
public void WriteLineNoTabs(
   System.String s
);
```

Public Fields

```
public static System.String DefaultTabString;
```

Protected Methods

```
protected void OutputTabs();
```

LanguageOptions Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum LanguageOptions
```

Members

CaseInsensitive

None

TempFileCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TempFileCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable, System.IDisposa
ble
```

Public Constructors

```
public TempFileCollection();
public TempFileCollection(
   System.String tempDir
);
public TempFileCollection(
   System.String tempDir,
   boolean keepFiles
);
```

Public Properties

```
public System.String get_BasePath();
public int get_Count();
public boolean get_KeepFiles();
public void set_KeepFiles(
   boolean value
);
public System.String get_TempDir();
```

Public Methods

```
public System.String AddExtension(
   System.String fileExtension
);
public System.String AddExtension(
   System.String fileExtension,
   boolean keepFile
);
public void AddFile(
   System.String fileName,
   boolean keepFile
);
public void CopyTo(
   System.String[] fileNames,
   int start
);
public void Delete();
public System.Collections.IEnumerator GetEnumerator();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

```
protected void Finalize();
```

System.Collections Namespace (Visual J# Syntax)

Classes

ArrayList

BitArray

CaseInsensitiveComparer

CaseInsensitiveHashCodeProvider

CollectionBase

Comparer

DictionaryBase

Hashtable

Queue

ReadOnlyCollectionBase

SortedList

Stack

Interfaces

ICollection

IComparer

IDictionary

IDictionaryEnumerator

IEnumerable

IEnumerator

IHashCodeProvider

IList

Structures

DictionaryEntry

ArrayList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ArrayList
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable, System.ICloneable
```

Public Constructors

```
public ArrayList();
public ArrayList(
   int capacity
);
public ArrayList(
   System.Collections.ICollection c
);
```

Public Properties

```
public int get_Capacity();
public void set_Capacity(
   int value
);
public int get_Count();
public boolean get_IsFixedSize();
public boolean get_IsReadOnly();
public boolean get_IsSynchronized();
public System.Object get_Item(
   int index
);
public void set_Item(
   int index,
   System.Object value
);
public System.Object get_SyncRoot();
```

Public Methods

```
public static System.Collections.ArrayList Adapter(
   System.Collections.IList list
);
public int Add(
   System.Object value
);
public void AddRange(
   System.Collections.ICollection c
);
public int BinarySearch(
   System.Object value
);
public int BinarySearch(
   System.Object value,
   System.Collections.IComparer comparer
```

```
);
public int BinarySearch(
   int index,
   int count,
   System.Object value,
   System.Collections.IComparer comparer
);
public void Clear();
public System.Object Clone();
public boolean Contains(
   System.Object item
);
public void CopyTo(
   System.Array array
);
public void CopyTo(
   System.Array array,
   int arrayIndex
);
public void CopyTo(
   int index,
   System.Array array,
   int arrayIndex,
   int count
);
public static System.Collections.ArrayList FixedSize(
   System.Collections.ArrayList list
);
public static System.Collections.IList FixedSize(
   System.Collections.IList list
);
public System.Collections.IEnumerator GetEnumerator();
public System.Collections.IEnumerator GetEnumerator(
   int index,
   int count
);
public System.Collections.ArrayList GetRange(
   int index,
   int count
);
public int IndexOf(
   System.Object value
);
public int IndexOf(
   System.Object value,
   int startIndex
);
public int IndexOf(
   System.Object value,
   int startIndex,
   int count
);
public void Insert(
   int index,
   System.Object value
);
public void InsertRange(
   int index,
   System.Collections.ICollection c
);
public int LastIndexOf(
   System.Object value
);
public int LastIndexOf(
   System.Object value,
   int startIndex
);
public int LastIndexOf(
```

```
   System.Object value,
   int startIndex,
   int count
);
public static System.Collections.ArrayList ReadOnly(
   System.Collections.ArrayList list
);
public static System.Collections.IList ReadOnly(
   System.Collections.IList list
);
public void Remove(
   System.Object obj
);
public void RemoveAt(
   int index
);
public void RemoveRange(
   int index,
   int count
);
public static System.Collections.ArrayList Repeat(
   System.Object value,
   int count
);
public void Reverse();
public void Reverse(
   int index,
   int count
);
public void SetRange(
   int index,
   System.Collections.ICollection c
);
public void Sort();
public void Sort(
   System.Collections.IComparer comparer
);
public void Sort(
   int index,
   int count,
   System.Collections.IComparer comparer
);
public static System.Collections.ArrayList Synchronized(
   System.Collections.ArrayList list
);
public static System.Collections.IList Synchronized(
   System.Collections.IList list
);
public System.Object[] ToArray();
public System.Array ToArray(
   System.Type type
);
public void TrimToSize();
```

BitArray Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class BitArray
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable, System.ICloneab
le
```

Public Constructors

```
public BitArray(
   boolean[] values
);
public BitArray(
   ubyte[] bytes
);
public BitArray(
   int length
);
public BitArray(
   int[] values
);
public BitArray(
   System.Collections.BitArray bits
);
public BitArray(
   int length,
   boolean defaultValue
);
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public boolean get_Item(
   int index
);
public void set_Item(
   int index,
   boolean value
);
public int get_Length();
public void set_Length(
   int value
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public System.Collections.BitArray And(
   System.Collections.BitArray value
);
public final System.Object Clone();
```

```
public final void CopyTo(
   System.Array array,
   int index
);
public boolean Get(
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public System.Collections.BitArray Not();
public System.Collections.BitArray Or(
   System.Collections.BitArray value
);
public void Set(
   int index,
   boolean value
);
public void SetAll(
   boolean value
);
public System.Collections.BitArray Xor(
   System.Collections.BitArray value
);
```

CaseInsensitiveComparer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class CaseInsensitiveComparer
   extends System.Object
   implements System.Collections.IComparer
```

Public Constructors

```
public CaseInsensitiveComparer();
public CaseInsensitiveComparer(
   System.Globalization.CultureInfo culture
);
```

Public Properties

```
public static System.Collections.CaseInsensitiveComparer get_Default();
```

Public Methods

```
public final int Compare(
   System.Object a,
   System.Object b
);
```

CaseInsensitiveHashCodeProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class CaseInsensitiveHashCodeProvider
   extends System.Object
   implements System.Collections.IHashCodeProvider
```

Public Constructors

```
public CaseInsensitiveHashCodeProvider();
public CaseInsensitiveHashCodeProvider(
   System.Globalization.CultureInfo culture
);
```

Public Properties

```
public static System.Collections.CaseInsensitiveHashCodeProvider get_Default();
```

Public Methods

```
public final int GetHashCode(
   System.Object obj
);
```

CollectionBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class CollectionBase
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Properties

```
public final int get_Count();
```

Public Methods

```
public final void Clear();
public final System.Collections.IEnumerator GetEnumerator();
public final void RemoveAt(
   int index
);
```

Protected Constructors

```
protected CollectionBase();
```

Protected Properties

```
protected System.Collections.ArrayList get_InnerList();
protected System.Collections.IList get_List();
```

Protected Methods

```
protected void OnClear();
protected void OnClearComplete();
protected void OnInsert(
   int index,
   System.Object value
);
protected void OnInsertComplete(
   int index,
   System.Object value
);
protected void OnRemove(
   int index,
   System.Object value
);
protected void OnRemoveComplete(
   int index,
   System.Object value
);
protected void OnSet(
   int index,
```

```
   System.Object oldValue,
   System.Object newValue
);
protected void OnSetComplete(
   int index,
   System.Object oldValue,
   System.Object newValue
);
protected void OnValidate(
   System.Object value
);
```

Comparer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Comparer
   extends System.Object
   implements System.Collections.IComparer
```

Public Methods

```
public final int Compare(
   System.Object a,
   System.Object b
);
```

Public Fields

```
public static final System.Collections.Comparer Default;
```

DictionaryBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class DictionaryBase
   extends System.Object
   implements System.Collections.IDictionary, System.Collections.ICollection, System.Collecti
ons.IEnumerable
```

Public Properties

```
public final int get_Count();
```

Public Methods

```
public final void Clear();
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IDictionaryEnumerator GetEnumerator();
```

Protected Constructors

```
protected DictionaryBase();
```

Protected Properties

```
protected System.Collections.IDictionary get_Dictionary();
protected System.Collections.Hashtable get_InnerHashtable();
```

Protected Methods

```
protected void OnClear();
protected void OnClearComplete();
protected System.Object OnGet(
   System.Object key,
   System.Object currentValue
);
protected void OnInsert(
   System.Object key,
   System.Object value
);
protected void OnInsertComplete(
   System.Object key,
   System.Object value
);
protected void OnRemove(
   System.Object key,
   System.Object value
);
protected void OnRemoveComplete(
```

```
    System.Object key,
    System.Object value
);
protected void OnSet(
    System.Object key,
    System.Object oldValue,
    System.Object newValue
);
protected void OnSetComplete(
    System.Object key,
    System.Object oldValue,
    System.Object newValue
);
protected void OnValidate(
    System.Object key,
    System.Object value
);
```

DictionaryEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DictionaryEntry
   extends System.ValueType
```

Public Constructors

```
public DictionaryEntry(
   System.Object key,
   System.Object value
);
```

Public Properties

```
public System.Object get_Key();
public void set_Key(
   System.Object value
);
public System.Object get_Value();
public void set_Value(
   System.Object value
);
```

Hashtable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class Hashtable
   extends System.Object
   implements System.Collections.IDictionary, System.Collections.ICollection, System.Collecti
ons.IEnumerable, System.Runtime.Serialization.ISerializable, System.Runtime.Serialization.IDe
serializationCallback, System.ICloneable
```

Public Constructors

```
public Hashtable();
public Hashtable(
   int capacity
);
public Hashtable(
   System.Collections.IDictionary d
);
public Hashtable(
   int capacity,
   float loadFactor
);
public Hashtable(
   System.Collections.IDictionary d,
   float loadFactor
);
public Hashtable(
   System.Collections.IHashCodeProvider hcp,
   System.Collections.IComparer comparer
);
public Hashtable(
   int capacity,
   System.Collections.IHashCodeProvider hcp,
   System.Collections.IComparer comparer
);
public Hashtable(
   System.Collections.IDictionary d,
   System.Collections.IHashCodeProvider hcp,
   System.Collections.IComparer comparer
);
public Hashtable(
   int capacity,
   float loadFactor,
   System.Collections.IHashCodeProvider hcp,
   System.Collections.IComparer comparer
);
public Hashtable(
   System.Collections.IDictionary d,
   float loadFactor,
   System.Collections.IHashCodeProvider hcp,
   System.Collections.IComparer comparer
);
```

Public Properties

```
public int get_Count();
```

```
public boolean get_IsFixedSize();
public boolean get_IsReadOnly();
public boolean get_IsSynchronized();
public System.Object get_Item(
   System.Object key
);
public void set_Item(
   System.Object key,
   System.Object value
);
public System.Collections.ICollection get_Keys();
public System.Object get_SyncRoot();
public System.Collections.ICollection get_Values();
```

Public Methods

```
public void Add(
   System.Object key,
   System.Object value
);
public void Clear();
public System.Object Clone();
public boolean Contains(
   System.Object key
);
public boolean ContainsKey(
   System.Object key
);
public boolean ContainsValue(
   System.Object value
);
public void CopyTo(
   System.Array array,
   int arrayIndex
);
public System.Collections.IDictionaryEnumerator GetEnumerator();
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public void OnDeserialization(
   System.Object sender
);
public void Remove(
   System.Object key
);
public static System.Collections.Hashtable Synchronized(
   System.Collections.Hashtable table
);
```

Protected Constructors

```
protected Hashtable(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Properties

```
protected System.Collections.IComparer get_comparer();
protected void set_comparer(
   System.Collections.IComparer value
```

```
);
protected System.Collections.IHashCodeProvider get_hcp();
protected void set_hcp(
   System.Collections.IHashCodeProvider value
);
```

Protected Methods

```
protected int GetHash(
   System.Object key
);
protected boolean KeyEquals(
   System.Object item,
   System.Object key
);
```

ICollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICollection
   implements System.Collections.IEnumerable
```

Public Properties

```
int get_Count();
boolean get_IsSynchronized();
System.Object get_SyncRoot();
```

Public Methods

```
void CopyTo(
   System.Array array,
   int index
);
```

IComparer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IComparer
```

Public Methods

```
int Compare(
   System.Object x,
   System.Object y
);
```

IDictionary Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public interface IDictionary
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
boolean get_IsFixedSize();
boolean get_IsReadOnly();
System.Object get_Item(
   System.Object key
);
void set_Item(
   System.Object key,
   System.Object value
);
System.Collections.ICollection get_Keys();
System.Collections.ICollection get_Values();
```

Public Methods

```
void Add(
   System.Object key,
   System.Object value
);
void Clear();
boolean Contains(
   System.Object key
);
System.Collections.IDictionaryEnumerator GetEnumerator();
void Remove(
   System.Object key
);
```

IDictionaryEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDictionaryEnumerator
   implements System.Collections.IEnumerator
```

Public Properties

```
System.Collections.DictionaryEntry get_Entry();
System.Object get_Key();
System.Object get_Value();
```

IEnumerable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "496B0ABE-CDEE-11d3-88E8-
00902754C43A") */
public interface IEnumerable
```

Public Methods

```
/** @attribute System.Runtime.InteropServices.DispIdAttribute(Value = -4) */
System.Collections.IEnumerator GetEnumerator();
```

IEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "496B0ABF-CDEE-11d3-88E8-
00902754C43A") */
public interface IEnumerator
```

Public Properties

```
System.Object get_Current();
```

Public Methods

```
boolean MoveNext();
void Reset();
```

IHashCodeProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IHashCodeProvider
```

Public Methods

```
int GetHashCode(
   System.Object obj
);
```

IList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public interface IList
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
boolean get_IsFixedSize();
boolean get_IsReadOnly();
System.Object get_Item(
   int index
);
void set_Item(
   int index,
   System.Object value
);
```

Public Methods

```
int Add(
   System.Object value
);
void Clear();
boolean Contains(
   System.Object value
);
int IndexOf(
   System.Object value
);
void Insert(
   int index,
   System.Object value
);
void Remove(
   System.Object value
);
void RemoveAt(
   int index
);
```

Queue Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Queue
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable, System.ICloneab
le
```

Public Constructors

```
public Queue();
public Queue(
   int capacity
);
public Queue(
   System.Collections.ICollection col
);
public Queue(
   int capacity,
   float growFactor
);
```

Public Properties

```
public int get_Count();
public boolean get_IsSynchronized();
public System.Object get_SyncRoot();
```

Public Methods

```
public void Clear();
public System.Object Clone();
public boolean Contains(
   System.Object obj
);
public void CopyTo(
   System.Array array,
   int index
);
public System.Object Dequeue();
public void Enqueue(
   System.Object obj
);
public System.Collections.IEnumerator GetEnumerator();
public System.Object Peek();
public static System.Collections.Queue Synchronized(
   System.Collections.Queue queue
);
public System.Object[] ToArray();
public void TrimToSize();
```

ReadOnlyCollectionBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class ReadOnlyCollectionBase
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
```

Public Methods

```
public final System.Collections.IEnumerator GetEnumerator();
```

Protected Constructors

```
protected ReadOnlyCollectionBase();
```

Protected Properties

```
protected System.Collections.ArrayList get_InnerList();
```

SortedList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SortedList
   extends System.Object
   implements System.Collections.IDictionary, System.Collections.ICollection, System.Collecti
ons.IEnumerable, System.ICloneable
```

Public Constructors

```
public SortedList();
public SortedList(
   int initialCapacity
);
public SortedList(
   System.Collections.IComparer comparer
);
public SortedList(
   System.Collections.IDictionary d
);
public SortedList(
   System.Collections.IComparer comparer,
   int capacity
);
public SortedList(
   System.Collections.IDictionary d,
   System.Collections.IComparer comparer
);
```

Public Properties

```
public int get_Capacity();
public void set_Capacity(
   int value
);
public int get_Count();
public boolean get_IsFixedSize();
public boolean get_IsReadOnly();
public boolean get_IsSynchronized();
public System.Object get_Item(
   System.Object key
);
public void set_Item(
   System.Object key,
   System.Object value
);
public System.Collections.ICollection get_Keys();
public System.Object get_SyncRoot();
public System.Collections.ICollection get_Values();
```

Public Methods

```
public void Add(
   System.Object key,
```

```
   System.Object value
);
public void Clear();
public System.Object Clone();
public boolean Contains(
   System.Object key
);
public boolean ContainsKey(
   System.Object key
);
public boolean ContainsValue(
   System.Object value
);
public void CopyTo(
   System.Array array,
   int arrayIndex
);
public System.Object GetByIndex(
   int index
);
public System.Collections.IDictionaryEnumerator GetEnumerator();
public System.Object GetKey(
   int index
);
public System.Collections.IList GetKeyList();
public System.Collections.IList GetValueList();
public int IndexOfKey(
   System.Object key
);
public int IndexOfValue(
   System.Object value
);
public void Remove(
   System.Object key
);
public void RemoveAt(
   int index
);
public void SetByIndex(
   int index,
   System.Object value
);
public static System.Collections.SortedList Synchronized(
   System.Collections.SortedList list
);
public void TrimToSize();
```

Stack Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Stack
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable, System.ICloneab
le
```

Public Constructors

```
public Stack();
public Stack(
   int initialCapacity
);
public Stack(
   System.Collections.ICollection col
);
```

Public Properties

```
public int get_Count();
public boolean get_IsSynchronized();
public System.Object get_SyncRoot();
```

Public Methods

```
public void Clear();
public System.Object Clone();
public boolean Contains(
   System.Object obj
);
public void CopyTo(
   System.Array array,
   int index
);
public System.Collections.IEnumerator GetEnumerator();
public System.Object Peek();
public System.Object Pop();
public void Push(
   System.Object obj
);
public static System.Collections.Stack Synchronized(
   System.Collections.Stack stack
);
public System.Object[] ToArray();
```

System.Collections.Specialized Namespace (Visual J# Syntax)

Classes

CollectionsUtil

HybridDictionary

ListDictionary

NameObjectCollectionBase

NameObjectCollectionBase.KeysCollection

NameValueCollection

StringCollection

StringDictionary

StringEnumerator

Structures

BitVector32

BitVector32.Section

BitVector32 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class BitVector32
   extends System.ValueType
```

Public Constructors

```
public BitVector32(
   int data
);
public BitVector32(
   System.Collections.Specialized.BitVector32 value
);
```

Public Properties

```
public int get_Data();
public boolean get_Item(
   int bit
);
public int get_Item(
   System.Collections.Specialized.BitVector32+Section section
);
public void set_Item(
   int bit,
   boolean value
);
public void set_Item(
   System.Collections.Specialized.BitVector32+Section section,
   int value
);
```

Public Methods

```
public static int CreateMask();
public static int CreateMask(
   int previous
);
public static System.Collections.Specialized.BitVector32+Section CreateSection(
   short maxValue
);
public static System.Collections.Specialized.BitVector32+Section CreateSection(
   short maxValue,
   System.Collections.Specialized.BitVector32+Section previous
);
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public System.String ToString();
public static System.String ToString(
   System.Collections.Specialized.BitVector32 value
);
```

BitVector32.Section Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Section
   extends System.ValueType
```

Public Properties

```
public short get_Mask();
public short get_Offset();
```

Public Methods

```
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public System.String ToString();
public static System.String ToString(
   System.Collections.Specialized.BitVector32+Section value
);
```

CollectionsUtil Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CollectionsUtil
   extends System.Object
```

Public Constructors

```
public CollectionsUtil();
```

Public Methods

```
public static System.Collections.Hashtable CreateCaseInsensitiveHashtable();
public static System.Collections.Hashtable CreateCaseInsensitiveHashtable(
   int capacity
);
public static System.Collections.Hashtable CreateCaseInsensitiveHashtable(
   System.Collections.IDictionary d
);
public static System.Collections.SortedList CreateCaseInsensitiveSortedList();
```

HybridDictionary Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HybridDictionary
   extends System.Object
   implements System.Collections.IDictionary, System.Collections.ICollection, System.Collecti
ons.IEnumerable
```

Public Constructors

```
public HybridDictionary();
public HybridDictionary(
   boolean caseInsensitive
);
public HybridDictionary(
   int initialSize
);
public HybridDictionary(
   int initialSize,
   boolean caseInsensitive
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsFixedSize();
public final boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public final System.Object get_Item(
   System.Object key
);
public final void set_Item(
   System.Object key,
   System.Object value
);
public final System.Collections.ICollection get_Keys();
public final System.Object get_SyncRoot();
public final System.Collections.ICollection get_Values();
```

Public Methods

```
public final void Add(
   System.Object key,
   System.Object value
);
public final void Clear();
public final boolean Contains(
   System.Object key
);
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IDictionaryEnumerator GetEnumerator();
```

```
public final void Remove(
   System.Object key
);
```

ListDictionary Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ListDictionary
   extends System.Object
   implements System.Collections.IDictionary, System.Collections.ICollection, System.Collecti
ons.IEnumerable
```

Public Constructors

```
public ListDictionary();
public ListDictionary(
   System.Collections.IComparer comparer
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsFixedSize();
public final boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public final System.Object get_Item(
   System.Object key
);
public final void set_Item(
   System.Object key,
   System.Object value
);
public final System.Collections.ICollection get_Keys();
public final System.Object get_SyncRoot();
public final System.Collections.ICollection get_Values();
```

Public Methods

```
public final void Add(
   System.Object key,
   System.Object value
);
public final void Clear();
public final boolean Contains(
   System.Object key
);
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IDictionaryEnumerator GetEnumerator();
public final void Remove(
   System.Object key
);
```

NameObjectCollectionBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class NameObjectCollectionBase
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable, System.Runtime.
Serialization.ISerializable, System.Runtime.Serialization.IDeserializationCallback
```

Public Properties

```
public int get_Count();
public System.Collections.Specialized.NameObjectCollectionBase+KeysCollection get_Keys();
```

Public Methods

```
public final System.Collections.IEnumerator GetEnumerator();
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public void OnDeserialization(
   System.Object sender
);
```

Protected Constructors

```
protected NameObjectCollectionBase();
protected NameObjectCollectionBase(
   int capacity
);
protected NameObjectCollectionBase(
   System.Collections.IHashCodeProvider hashProvider,
   System.Collections.IComparer comparer
);
protected NameObjectCollectionBase(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
protected NameObjectCollectionBase(
   int capacity,
   System.Collections.IHashCodeProvider hashProvider,
   System.Collections.IComparer comparer
);
```

Protected Properties

```
protected boolean get_IsReadOnly();
protected void set_IsReadOnly(
   boolean value
);
```

Protected Methods

```
protected void BaseAdd(
   System.String name,
   System.Object value
);
protected void BaseClear();
protected System.Object BaseGet(
   int index
);
protected System.Object BaseGet(
   System.String name
);
protected System.String[] BaseGetAllKeys();
protected System.Object[] BaseGetAllValues();
protected System.Object[] BaseGetAllValues(
   System.Type type
);
protected System.String BaseGetKey(
   int index
);
protected boolean BaseHasKeys();
protected void BaseRemove(
   System.String name
);
protected void BaseRemoveAt(
   int index
);
protected void BaseSet(
   int index,
   System.Object value
);
protected void BaseSet(
   System.String name,
   System.Object value
);
```

NameObjectCollectionBase.KeysCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class KeysCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public System.String get_Item(
   int index
);
```

Public Methods

```
public System.String Get(
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
```

NameValueCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class NameValueCollection
   extends System.Collections.Specialized.NameObjectCollectionBase
```

Public Constructors

```
public NameValueCollection();
public NameValueCollection(
   int capacity
);
public NameValueCollection(
   System.Collections.Specialized.NameValueCollection col
);
public NameValueCollection(
   int capacity,
   System.Collections.Specialized.NameValueCollection col
);
public NameValueCollection(
   System.Collections.IHashCodeProvider hashProvider,
   System.Collections.IComparer comparer
);
public NameValueCollection(
   int capacity,
   System.Collections.IHashCodeProvider hashProvider,
   System.Collections.IComparer comparer
);
```

Public Properties

```
public System.String[] get_AllKeys();
public System.String get_Item(
   int index
);
public System.String get_Item(
   System.String name
);
public void set_Item(
   System.String name,
   System.String value
);
```

Public Methods

```
public void Add(
   System.Collections.Specialized.NameValueCollection c
);
public void Add(
   System.String name,
   System.String value
);
public void Clear();
public void CopyTo(
```

```
   System.Array dest,
   int index
);
public System.String Get(
   int index
);
public System.String Get(
   System.String name
);
public System.String GetKey(
   int index
);
public System.String[] GetValues(
   int index
);
public System.String[] GetValues(
   System.String name
);
public boolean HasKeys();
public void Remove(
   System.String name
);
public void Set(
   System.String name,
   System.String value
);
```

Protected Constructors

```
protected NameValueCollection(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Methods

```
protected void InvalidateCachedArrays();
```

StringCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class StringCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public StringCollection();
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.String get_Item(
   int index
);
public void set_Item(
   int index,
   System.String value
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public int Add(
   System.String value
);
public void AddRange(
   System.String[] value
);
public final void Clear();
public boolean Contains(
   System.String value
);
public void CopyTo(
   System.String[] array,
   int index
);
public System.Collections.Specialized.StringEnumerator GetEnumerator();
public int IndexOf(
   System.String value
);
public void Insert(
   int index,
   System.String value
);
public void Remove(
   System.String value
);
public final void RemoveAt(
```

```
    int index
);
```

StringDictionary Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.Design.Serialization.DesignerSerializerAttribute(Seriali
zerTypeName = "System.Diagnostics.Design.StringDictionaryCodeDomSerializer, System.Design, Ve
rsion=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a", SerializerBaseTypeName =
"System.ComponentModel.Design.Serialization.CodeDomSerializer, System.Design, Version=1.0.330
0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class StringDictionary
   extends System.Object
   implements System.Collections.IEnumerable
```

Public Constructors

```
public StringDictionary();
```

Public Properties

```
public int get_Count();
public boolean get_IsSynchronized();
public System.String get_Item(
   System.String key
);
public void set_Item(
   System.String key,
   System.String value
);
public System.Collections.ICollection get_Keys();
public System.Object get_SyncRoot();
public System.Collections.ICollection get_Values();
```

Public Methods

```
public void Add(
   System.String key,
   System.String value
);
public void Clear();
public boolean ContainsKey(
   System.String key
);
public boolean ContainsValue(
   System.String value
);
public void CopyTo(
   System.Array array,
   int index
);
public System.Collections.IEnumerator GetEnumerator();
public void Remove(
   System.String key
);
```

StringEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class StringEnumerator
   extends System.Object
```

Public Properties

```
public System.String get_Current();
```

Public Methods

```
public boolean MoveNext();
public void Reset();
```

System.ComponentModel Namespace (Visual J# Syntax)

Classes

AmbientValueAttribute

ArrayConverter

AttributeCollection

BaseNumberConverter

BindableAttribute

BooleanConverter

BrowsableAttribute

ByteConverter

CancelEventArgs

CategoryAttribute

CharConverter

CollectionChangeEventArgs

CollectionConverter

Component

ComponentCollection

ComponentConverter

ComponentEditor

Container

CultureInfoConverter

DateTimeConverter

DecimalConverter

DefaultEventAttribute

DefaultPropertyAttribute

DefaultValueAttribute

DescriptionAttribute

DesignerAttribute

DesignerCategoryAttribute

DesignerSerializationVisibilityAttribute

DesignOnlyAttribute

DesignTimeVisibleAttribute

DoubleConverter

EditorAttribute

EditorBrowsableAttribute

EnumConverter

EventDescriptor

EventDescriptorCollection

EventHandlerList

ExpandableObjectConverter

ExtenderProvidedPropertyAttribute

GuidConverter

ImmutableObjectAttribute

InheritanceAttribute

InstallerTypeAttribute

Int16Converter

Int32Converter

Int64Converter

InvalidEnumArgumentException

License

LicenseContext

LicenseException

LicenseManager

LicenseProvider

LicenseProviderAttribute

LicFileLicenseProvider

ListBindableAttribute

ListChangedEventArgs

LocalizableAttribute

MarshalByValueComponent

MemberDescriptor

MergablePropertyAttribute

NotifyParentPropertyAttribute

ParenthesizePropertyNameAttribute

PropertyChangedEventArgs

PropertyDescriptor

PropertyDescriptorCollection

PropertyTabAttribute

ProvidePropertyAttribute

ReadOnlyAttribute

RecommendedAsConfigurableAttribute

ReferenceConverter

RefreshEventArgs

RefreshPropertiesAttribute

RunInstallerAttribute

SByteConverter

SingleConverter

StringConverter

SyntaxCheck

TimeSpanConverter

ToolboxItemAttribute

ToolboxItemFilterAttribute

TypeConverter

TypeConverter.SimplePropertyDescriptor

TypeConverter.StandardValuesCollection

TypeConverterAttribute

TypeDescriptor

TypeListConverter

UInt16Converter

UInt32Converter

UInt64Converter

WarningException

Win32Exception

Interfaces

IBindingList

IComNativeDescriptorHandler

IComponent

IContainer

ICustomTypeDescriptor

IDataErrorInfo

IEditableObject

IExtenderProvider

IListSource

ISite

ISupportInitialize

ISynchronizeInvoke

ITypeDescriptorContext

ITypedList

Delegates

CancelEventHandler

CollectionChangeEventHandler

ListChangedEventHandler

PropertyChangedEventHandler

RefreshEventHandler

Enumerations

BindableSupport

CollectionChangeAction

DesignerSerializationVisibility

EditorBrowsableState

InheritanceLevel

LicenseUsageMode

ListChangedType

ListSortDirection

PropertyTabScope

RefreshProperties

ToolboxItemFilterType

AmbientValueAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class AmbientValueAttribute
   extends System.Attribute
```

Public Constructors

```
public AmbientValueAttribute(
   boolean value
);
public AmbientValueAttribute(
   ubyte value
);
public AmbientValueAttribute(
   char value
);
public AmbientValueAttribute(
   double value
);
public AmbientValueAttribute(
   short value
);
public AmbientValueAttribute(
   int value
);
public AmbientValueAttribute(
   long value
);
public AmbientValueAttribute(
   float value
);
public AmbientValueAttribute(
   System.Object value
);
public AmbientValueAttribute(
   System.String value
);
public AmbientValueAttribute(
   System.Type type,
   System.String value
);
```

Public Properties

```
public System.Object get_Value();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

ArrayConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ArrayConverter
   extends System.ComponentModel.CollectionConverter
```

Public Constructors

```
public ArrayConverter();
```

Public Methods

```
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value,
   System.Attribute[] attributes
);
public boolean GetPropertiesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

AttributeCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class AttributeCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public AttributeCollection(
   System.Attribute[] attributes
);
```

Public Properties

```
public int get_Count();
public System.Attribute get_Item(
   int index
);
public System.Attribute get_Item(
   System.Type attributeType
);
```

Public Methods

```
public boolean Contains(
   System.Attribute attribute
);
public boolean Contains(
   System.Attribute[] attributes
);
public final void CopyTo(
   System.Array array,
   int index
);
public System.Collections.IEnumerator GetEnumerator();
public boolean Matches(
   System.Attribute attribute
);
public boolean Matches(
   System.Attribute[] attributes
);
```

Public Fields

```
public static final System.ComponentModel.AttributeCollection Empty;
```

Protected Methods

```
protected System.Attribute GetDefaultAttribute(
```

```
    System.Type attributeType
);
```

BaseNumberConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class BaseNumberConverter
   extends System.ComponentModel.TypeConverter
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type t
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

Protected Constructors

```
protected BaseNumberConverter();
```

BindableAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class BindableAttribute
   extends System.Attribute
```

Public Constructors

```
public BindableAttribute(
   boolean bindable
);
public BindableAttribute(
   System.ComponentModel.BindableSupport flags
);
```

Public Properties

```
public boolean get_Bindable();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.BindableAttribute Default;
public static final System.ComponentModel.BindableAttribute No;
public static final System.ComponentModel.BindableAttribute Yes;
```

BindableSupport Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum BindableSupport
```

Members

Default

No

Yes

BooleanConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BooleanConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public BooleanConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

BrowsableAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class BrowsableAttribute
   extends System.Attribute
```

Public Constructors

```
public BrowsableAttribute(
   boolean browsable
);
```

Public Properties

```
public boolean get_Browsable();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.BrowsableAttribute Default;
public static final System.ComponentModel.BrowsableAttribute No;
public static final System.ComponentModel.BrowsableAttribute Yes;
```

ByteConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ByteConverter
   extends System.ComponentModel.BaseNumberConverter
```

Public Constructors

```
public ByteConverter();
```

CancelEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CancelEventArgs
   extends System.EventArgs
```

Public Constructors

```
public CancelEventArgs();
public CancelEventArgs(
   boolean cancel
);
```

Public Properties

```
public boolean get_Cancel();
public void set_Cancel(
   boolean value
);
```

CancelEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void CancelEventHandler(
   System.Object sender,
   System.ComponentModel.CancelEventArgs e
);
```

CategoryAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class CategoryAttribute
   extends System.Attribute
```

Public Constructors

```
public CategoryAttribute();
public CategoryAttribute(
   System.String category
);
```

Public Properties

```
public static System.ComponentModel.CategoryAttribute get_Action();
public static System.ComponentModel.CategoryAttribute get_Appearance();
public static System.ComponentModel.CategoryAttribute get_Behavior();
public System.String get_Category();
public static System.ComponentModel.CategoryAttribute get_Data();
public static System.ComponentModel.CategoryAttribute get_Default();
public static System.ComponentModel.CategoryAttribute get_Design();
public static System.ComponentModel.CategoryAttribute get_DragDrop();
public static System.ComponentModel.CategoryAttribute get_Focus();
public static System.ComponentModel.CategoryAttribute get_Format();
public static System.ComponentModel.CategoryAttribute get_Key();
public static System.ComponentModel.CategoryAttribute get_Layout();
public static System.ComponentModel.CategoryAttribute get_Mouse();
public static System.ComponentModel.CategoryAttribute get_WindowStyle();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Protected Methods

```
protected System.String GetLocalizedString(
   System.String value
);
```

CharConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CharConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public CharConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

CollectionChangeAction Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CollectionChangeAction
```

Members

Add

Refresh

Remove

CollectionChangeEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CollectionChangeEventArgs
   extends System.EventArgs
```

Public Constructors

```
public CollectionChangeEventArgs(
   System.ComponentModel.CollectionChangeAction action,
   System.Object element
);
```

Public Properties

```
public System.ComponentModel.CollectionChangeAction get_Action();
public System.Object get_Element();
```

CollectionChangeEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void CollectionChangeEventHandler(
   System.Object sender,
   System.ComponentModel.CollectionChangeEventArgs e
);
```

CollectionConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CollectionConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public CollectionConverter();
```

Public Methods

```
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value,
   System.Attribute[] attributes
);
public boolean GetPropertiesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

Component Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerCategoryAttribute(Category = "Component") */
public class Component
   extends System.MarshalByRefObject
   implements System.ComponentModel.IComponent, System.IDisposable
```

Public Constructors

```
public Component();
```

Public Properties

```
public System.ComponentModel.IContainer get_Container();
public System.ComponentModel.ISite get_Site();
public void set_Site(
   System.ComponentModel.ISite value
);
```

Public Methods

```
public final void Dispose();
public System.String ToString();
```

Public Events

```
public final void add_Disposed(
   System.EventHandler value
);
public final void remove_Disposed(
   System.EventHandler value
);
```

Protected Properties

```
protected boolean get_DesignMode();
protected System.ComponentModel.EventHandlerList get_Events();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
protected System.Object GetService(
   System.Type service
);
```

ComponentCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class ComponentCollection
   extends System.Collections.ReadOnlyCollectionBase
```

Public Constructors

```
public ComponentCollection(
   System.ComponentModel.IComponent[] components
);
```

Public Properties

```
public System.ComponentModel.IComponent get_Item(
   int index
);
public System.ComponentModel.IComponent get_Item(
   System.String name
);
```

Public Methods

```
public void CopyTo(
   System.ComponentModel.IComponent[] array,
   int index
);
```

ComponentConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ComponentConverter
   extends System.ComponentModel.ReferenceConverter
```

Public Constructors

```
public ComponentConverter(
   System.Type type
);
```

Public Methods

```
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value,
   System.Attribute[] attributes
);
public boolean GetPropertiesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

ComponentEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class ComponentEditor
   extends System.Object
```

Public Methods

```
public boolean EditComponent(
   System.Object component
);
public abstract boolean EditComponent(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object component
);
```

Protected Constructors

```
protected ComponentEditor();
```

Container Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Container
   extends System.Object
   implements System.ComponentModel.IContainer, System.IDisposable
```

Public Constructors

```
public Container();
```

Public Properties

```
public System.ComponentModel.ComponentCollection get_Components();
```

Public Methods

```
public void Add(
   System.ComponentModel.IComponent component
);
public void Add(
   System.ComponentModel.IComponent component,
   System.String name
);
public final void Dispose();
public void Remove(
   System.ComponentModel.IComponent component
);
```

Protected Methods

```
protected System.ComponentModel.ISite CreateSite(
   System.ComponentModel.IComponent component,
   System.String name
);
protected void Dispose(
   boolean disposing
);
protected void Finalize();
protected System.Object GetService(
   System.Type service
);
```

CultureInfoConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CultureInfoConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public CultureInfoConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

DateTimeConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DateTimeConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public DateTimeConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

DecimalConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DecimalConverter
   extends System.ComponentModel.BaseNumberConverter
```

Public Constructors

```
public DecimalConverter();
```

Public Methods

```
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

DefaultEventAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class DefaultEventAttribute
   extends System.Attribute
```

Public Constructors

```
public DefaultEventAttribute(
   System.String name
);
```

Public Properties

```
public System.String get_Name();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

Public Fields

```
public static final System.ComponentModel.DefaultEventAttribute Default;
```

DefaultPropertyAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class DefaultPropertyAttribute
   extends System.Attribute
```

Public Constructors

```
public DefaultPropertyAttribute(
   System.String name
);
```

Public Properties

```
public System.String get_Name();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

Public Fields

```
public static final System.ComponentModel.DefaultPropertyAttribute Default;
```

DefaultValueAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class DefaultValueAttribute
   extends System.Attribute
```

Public Constructors

```
public DefaultValueAttribute(
   boolean value
);
public DefaultValueAttribute(
   ubyte value
);
public DefaultValueAttribute(
   char value
);
public DefaultValueAttribute(
   double value
);
public DefaultValueAttribute(
   short value
);
public DefaultValueAttribute(
   int value
);
public DefaultValueAttribute(
   long value
);
public DefaultValueAttribute(
   float value
);
public DefaultValueAttribute(
   System.Object value
);
public DefaultValueAttribute(
   System.String value
);
public DefaultValueAttribute(
   System.Type type,
   System.String value
);
```

Public Properties

```
public System.Object get_Value();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

DescriptionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class DescriptionAttribute
   extends System.Attribute
```

Public Constructors

```
public DescriptionAttribute();
public DescriptionAttribute(
   System.String description
);
```

Public Properties

```
public System.String get_Description();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

Public Fields

```
public static final System.ComponentModel.DescriptionAttribute Default;
```

Protected Properties

```
protected System.String get_DescriptionValue();
protected void set_DescriptionValue(
   System.String value
);
```

DesignerAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Interface, AllowMultiple = tru
e, Inherited = true) */
public class DesignerAttribute
   extends System.Attribute
```

Public Constructors

```
public DesignerAttribute(
   System.String designerTypeName
);
public DesignerAttribute(
   System.Type designerType
);
public DesignerAttribute(
   System.String designerTypeName,
   System.String designerBaseTypeName
);
public DesignerAttribute(
   System.String designerTypeName,
   System.Type designerBaseType
);
public DesignerAttribute(
   System.Type designerType,
   System.Type designerBaseType
);
```

Public Properties

```
public System.String get_DesignerBaseTypeName();
public System.String get_DesignerTypeName();
public System.Object get_TypeId();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

DesignerCategoryAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class DesignerCategoryAttribute
   extends System.Attribute
```

Public Constructors

```
public DesignerCategoryAttribute();
public DesignerCategoryAttribute(
   System.String category
);
```

Public Properties

```
public System.String get_Category();
public System.Object get_TypeId();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.DesignerCategoryAttribute Component;
public static final System.ComponentModel.DesignerCategoryAttribute Default;
public static final System.ComponentModel.DesignerCategoryAttribute Form;
public static final System.ComponentModel.DesignerCategoryAttribute Generic;
```

DesignerSerializationVisibility Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum DesignerSerializationVisibility
```

Members

Content

Hidden

Visible

DesignerSerializationVisibilityAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, Property, AllowMultiple = fal
se, Inherited = true) */
public class DesignerSerializationVisibilityAttribute
   extends System.Attribute
```

Public Constructors

```
public DesignerSerializationVisibilityAttribute(
   System.ComponentModel.DesignerSerializationVisibility visibility
);
```

Public Properties

```
public System.ComponentModel.DesignerSerializationVisibility get_Visibility();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.DesignerSerializationVisibilityAttribute Content;
public static final System.ComponentModel.DesignerSerializationVisibilityAttribute Default;
public static final System.ComponentModel.DesignerSerializationVisibilityAttribute Hidden;
public static final System.ComponentModel.DesignerSerializationVisibilityAttribute Visible;
```

DesignOnlyAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class DesignOnlyAttribute
   extends System.Attribute
```

Public Constructors

```
public DesignOnlyAttribute(
   boolean isDesignOnly
);
```

Public Properties

```
public boolean get_IsDesignOnly();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.DesignOnlyAttribute Default;
public static final System.ComponentModel.DesignOnlyAttribute No;
public static final System.ComponentModel.DesignOnlyAttribute Yes;
```

DesignTimeVisibleAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Interface, AllowMultiple = fal
se, Inherited = true) */
public class DesignTimeVisibleAttribute
   extends System.Attribute
```

Public Constructors

```
public DesignTimeVisibleAttribute();
public DesignTimeVisibleAttribute(
   boolean visible
);
```

Public Properties

```
public boolean get_Visible();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.DesignTimeVisibleAttribute Default;
public static final System.ComponentModel.DesignTimeVisibleAttribute No;
public static final System.ComponentModel.DesignTimeVisibleAttribute Yes;
```

DoubleConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DoubleConverter
   extends System.ComponentModel.BaseNumberConverter
```

Public Constructors

```
public DoubleConverter();
```

EditorAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = true, Inherited
= true) */
public class EditorAttribute
   extends System.Attribute
```

Public Constructors

```
public EditorAttribute();
public EditorAttribute(
   System.String typeName,
   System.String baseTypeName
);
public EditorAttribute(
   System.String typeName,
   System.Type baseType
);
public EditorAttribute(
   System.Type type,
   System.Type baseType
);
```

Public Properties

```
public System.String get_EditorBaseTypeName();
public System.String get_EditorTypeName();
public System.Object get_TypeId();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

EditorBrowsableAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, Enum, Constructor, Met
hod, Property, Field, Event, Interface, Delegate, AllowMultiple = false, Inherited = true) */
public class EditorBrowsableAttribute
   extends System.Attribute
```

Public Constructors

```
public EditorBrowsableAttribute();
public EditorBrowsableAttribute(
   System.ComponentModel.EditorBrowsableState state
);
```

Public Properties

```
public System.ComponentModel.EditorBrowsableState get_State();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

EditorBrowsableState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum EditorBrowsableState
```

Members

Advanced

Always

Never

EnumConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EnumConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public EnumConverter(
   System.Type type
);
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean IsValid(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value
);
```

Protected Properties

```
protected System.Collections.IComparer get_Comparer();
protected System.Type get_EnumType();
protected System.ComponentModel.TypeConverter+StandardValuesCollection get_Values();
protected void set_Values(
   System.ComponentModel.TypeConverter+StandardValuesCollection value
);
```

EventDescriptor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public abstract class EventDescriptor
   extends System.ComponentModel.MemberDescriptor
```

Public Properties

```
public abstract System.Type get_ComponentType();
public abstract System.Type get_EventType();
public abstract boolean get_IsMulticast();
```

Public Methods

```
public abstract void AddEventHandler(
   System.Object component,
   System.Delegate value
);
public abstract void RemoveEventHandler(
   System.Object component,
   System.Delegate value
);
```

Protected Constructors

```
protected EventDescriptor(
   System.ComponentModel.MemberDescriptor descr
);
protected EventDescriptor(
   System.ComponentModel.MemberDescriptor descr,
   System.Attribute[] attrs
);
protected EventDescriptor(
   System.String name,
   System.Attribute[] attrs
);
```

EventDescriptorCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class EventDescriptorCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public EventDescriptorCollection(
   System.ComponentModel.EventDescriptor[] events
);
```

Public Properties

```
public int get_Count();
public System.ComponentModel.EventDescriptor get_Item(
   int index
);
public System.ComponentModel.EventDescriptor get_Item(
   System.String name
);
```

Public Methods

```
public int Add(
   System.ComponentModel.EventDescriptor value
);
public void Clear();
public boolean Contains(
   System.ComponentModel.EventDescriptor value
);
public System.ComponentModel.EventDescriptor Find(
   System.String name,
   boolean ignoreCase
);
public System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.ComponentModel.EventDescriptor value
);
public void Insert(
   int index,
   System.ComponentModel.EventDescriptor value
);
public void Remove(
   System.ComponentModel.EventDescriptor value
);
public void RemoveAt(
   int index
);
public System.ComponentModel.EventDescriptorCollection Sort();
public System.ComponentModel.EventDescriptorCollection Sort(
   System.Collections.IComparer comparer
```

```
);
public System.ComponentModel.EventDescriptorCollection Sort(
   System.String[] names
);
public System.ComponentModel.EventDescriptorCollection Sort(
   System.String[] names,
   System.Collections.IComparer comparer
);
```

Public Fields

```
public static final System.ComponentModel.EventDescriptorCollection Empty;
```

Protected Methods

```
protected void InternalSort(
   System.Collections.IComparer sorter
);
protected void InternalSort(
   System.String[] names
);
```

EventHandlerList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class EventHandlerList
   extends System.Object
   implements System.IDisposable
```

Public Constructors

```
public EventHandlerList();
```

Public Properties

```
public System.Delegate get_Item(
   System.Object key
);
public void set_Item(
   System.Object key,
   System.Delegate value
);
```

Public Methods

```
public void AddHandler(
   System.Object key,
   System.Delegate value
);
public final void Dispose();
public void RemoveHandler(
   System.Object key,
   System.Delegate value
);
```

ExpandableObjectConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ExpandableObjectConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public ExpandableObjectConverter();
```

Public Methods

```
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value,
   System.Attribute[] attributes
);
public boolean GetPropertiesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

ExtenderProvidedPropertyAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class ExtenderProvidedPropertyAttribute
   extends System.Attribute
```

Public Constructors

```
public ExtenderProvidedPropertyAttribute();
```

Public Properties

```
public System.ComponentModel.PropertyDescriptor get_ExtenderProperty();
public System.ComponentModel.IExtenderProvider get_Provider();
public System.Type get_ReceiverType();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

GuidConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class GuidConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public GuidConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

IBindingList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IBindingList
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Properties

```
boolean get_AllowEdit();
boolean get_AllowNew();
boolean get_AllowRemove();
boolean get_IsSorted();
System.ComponentModel.ListSortDirection get_SortDirection();
System.ComponentModel.PropertyDescriptor get_SortProperty();
boolean get_SupportsChangeNotification();
boolean get_SupportsSearching();
boolean get_SupportsSorting();
```

Public Methods

```
void AddIndex(
   System.ComponentModel.PropertyDescriptor property
);
System.Object AddNew();
void ApplySort(
   System.ComponentModel.PropertyDescriptor property,
   System.ComponentModel.ListSortDirection direction
);
int Find(
   System.ComponentModel.PropertyDescriptor property,
   System.Object key
);
void RemoveIndex(
   System.ComponentModel.PropertyDescriptor property
);
void RemoveSort();
```

Public Events

```
void add_ListChanged(
   System.ComponentModel.ListChangedEventHandler value
);
void remove_ListChanged(
   System.ComponentModel.ListChangedEventHandler value
);
```

IComNativeDescriptorHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IComNativeDescriptorHandler
```

Public Methods

```
System.ComponentModel.AttributeCollection GetAttributes(
   System.Object component
);
System.String GetClassName(
   System.Object component
);
System.ComponentModel.TypeConverter GetConverter(
   System.Object component
);
System.ComponentModel.EventDescriptor GetDefaultEvent(
   System.Object component
);
System.ComponentModel.PropertyDescriptor GetDefaultProperty(
   System.Object component
);
System.Object GetEditor(
   System.Object component,
   System.Type baseEditorType
);
System.ComponentModel.EventDescriptorCollection GetEvents(
   System.Object component
);
System.ComponentModel.EventDescriptorCollection GetEvents(
   System.Object component,
   System.Attribute[] attributes
);
System.String GetName(
   System.Object component
);
System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.Object component,
   System.Attribute[] attributes
);
System.Object GetPropertyValue(
   System.Object component,
   int dispid,
   boolean success
);
System.Object GetPropertyValue(
   System.Object component,
   System.String propertyName,
   boolean success
);
```

IComponent Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IRootDesigner, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77
a5c561934e089", DesignerTypeName = "System.Windows.Forms.Design.ComponentDocumentDesigner, Sy
stem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.Design.Serialization.RootDesignerSerializerAttribute(Rel
oadable = true, SerializerTypeName = "System.ComponentModel.Design.Serialization.RootCodeDomS
erializer, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3
a", SerializerBaseTypeName = "System.ComponentModel.Design.Serialization.CodeDomSerializer, S
ystem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Compo
nentModel.ComponentConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77
a5c561934e089") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c5
61934e089", DesignerTypeName = "System.ComponentModel.Design.ComponentDesigner, System.Design
, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public interface IComponent
   implements System.IDisposable
```

Public Properties

```
System.ComponentModel.ISite get_Site();
void set_Site(
   System.ComponentModel.ISite value
);
```

Public Events

```
void add_Disposed(
   System.EventHandler value
);
void remove_Disposed(
   System.EventHandler value
);
```

IContainer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public interface IContainer
   implements System.IDisposable
```

Public Properties

```
System.ComponentModel.ComponentCollection get_Components();
```

Public Methods

```
void Add(
   System.ComponentModel.IComponent component
);
void Add(
   System.ComponentModel.IComponent component,
   System.String name
);
void Remove(
   System.ComponentModel.IComponent component
);
```

ICustomTypeDescriptor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICustomTypeDescriptor
```

Public Methods

```
System.ComponentModel.AttributeCollection GetAttributes();
System.String GetClassName();
System.String GetComponentName();
System.ComponentModel.TypeConverter GetConverter();
System.ComponentModel.EventDescriptor GetDefaultEvent();
System.ComponentModel.PropertyDescriptor GetDefaultProperty();
System.Object GetEditor(
   System.Type editorBaseType
);
System.ComponentModel.EventDescriptorCollection GetEvents();
System.ComponentModel.EventDescriptorCollection GetEvents(
   System.Attribute[] attributes
);
System.ComponentModel.PropertyDescriptorCollection GetProperties();
System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.Attribute[] attributes
);
System.Object GetPropertyOwner(
   System.ComponentModel.PropertyDescriptor pd
);
```

IDataErrorInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public interface IDataErrorInfo
```

Public Properties

```
System.String get_Error();
System.String get_Item(
   System.String columnName
);
```

IEditableObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IEditableObject
```

Public Methods

```
void BeginEdit();
void CancelEdit();
void EndEdit();
```

IExtenderProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IExtenderProvider
```

Public Methods

```
boolean CanExtend(
   System.Object extendee
);
```

IListSource Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IListSource
```

Public Properties

```
boolean get_ContainsListCollection();
```

Public Methods

```
System.Collections.IList GetList();
```

ImmutableObjectAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class ImmutableObjectAttribute
   extends System.Attribute
```

Public Constructors

```
public ImmutableObjectAttribute(
   boolean immutable
);
```

Public Properties

```
public boolean get_Immutable();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.ImmutableObjectAttribute Default;
public static final System.ComponentModel.ImmutableObjectAttribute No;
public static final System.ComponentModel.ImmutableObjectAttribute Yes;
```

InheritanceAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Event, AllowMultiple
= false, Inherited = true) */
public class InheritanceAttribute
   extends System.Attribute
```

Public Constructors

```
public InheritanceAttribute();
public InheritanceAttribute(
   System.ComponentModel.InheritanceLevel inheritanceLevel
);
```

Public Properties

```
public System.ComponentModel.InheritanceLevel get_InheritanceLevel();
```

Public Methods

```
public boolean Equals(
   System.Object value
);
public int GetHashCode();
public boolean IsDefaultAttribute();
public System.String ToString();
```

Public Fields

```
public static final System.ComponentModel.InheritanceAttribute Default;
public static final System.ComponentModel.InheritanceAttribute Inherited;
public static final System.ComponentModel.InheritanceAttribute InheritedReadOnly;
public static final System.ComponentModel.InheritanceAttribute NotInherited;
```

InheritanceLevel Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum InheritanceLevel
```

Members

Inherited

InheritedReadOnly

NotInherited

InstallerTypeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class InstallerTypeAttribute
   extends System.Attribute
```

Public Constructors

```
public InstallerTypeAttribute(
   System.String typeName
);
public InstallerTypeAttribute(
   System.Type installerType
);
```

Public Properties

```
public System.Type get_InstallerType();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

Int16Converter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Int16Converter
   extends System.ComponentModel.BaseNumberConverter
```

Public Constructors

```
public Int16Converter();
```

Int32Converter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Int32Converter
   extends System.ComponentModel.BaseNumberConverter
```

Public Constructors

```
public Int32Converter();
```

Int64Converter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Int64Converter
   extends System.ComponentModel.BaseNumberConverter
```

Public Constructors

```
public Int64Converter();
```

InvalidEnumArgumentException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InvalidEnumArgumentException
   extends System.ArgumentException
```

Public Constructors

```
public InvalidEnumArgumentException();
public InvalidEnumArgumentException(
   System.String message
);
public InvalidEnumArgumentException(
   System.String argumentName,
   int invalidValue,
   System.Type enumClass
);
```

ISite Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public interface ISite
   implements System.IServiceProvider
```

Public Properties

```
System.ComponentModel.IComponent get_Component();
System.ComponentModel.IContainer get_Container();
boolean get_DesignMode();
System.String get_Name();
void set_Name(
   System.String value
);
```

ISupportInitialize Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISupportInitialize
```

Public Methods

```
void BeginInit();
void EndInit();
```

ISynchronizeInvoke Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISynchronizeInvoke
```

Public Properties

```
boolean get_InvokeRequired();
```

Public Methods

```
System.IAsyncResult BeginInvoke(
   System.Delegate method,
   System.Object[] args
);
System.Object EndInvoke(
   System.IAsyncResult result
);
System.Object Invoke(
   System.Delegate method,
   System.Object[] args
);
```

ITypeDescriptorContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public interface ITypeDescriptorContext
   implements System.IServiceProvider
```

Public Properties

```
System.ComponentModel.IContainer get_Container();
System.Object get_Instance();
System.ComponentModel.PropertyDescriptor get_PropertyDescriptor();
```

Public Methods

```
void OnComponentChanged();
boolean OnComponentChanging();
```

ITypedList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ITypedList
```

Public Methods

```
System.ComponentModel.PropertyDescriptorCollection GetItemProperties(
   System.ComponentModel.PropertyDescriptor[] listAccessors
);
System.String GetListName(
   System.ComponentModel.PropertyDescriptor[] listAccessors
);
```

License Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class License
   extends System.Object
   implements System.IDisposable
```

Public Properties

```
public abstract System.String get_LicenseKey();
```

Public Methods

```
public abstract void Dispose();
```

Protected Constructors

```
protected License();
```

LicenseContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LicenseContext
   extends System.Object
   implements System.IServiceProvider
```

Public Constructors

```
public LicenseContext();
```

Public Properties

```
public System.ComponentModel.LicenseUsageMode get_UsageMode();
```

Public Methods

```
public System.String GetSavedLicenseKey(
   System.Type type,
   System.Reflection.Assembly resourceAssembly
);
public System.Object GetService(
   System.Type type
);
public void SetSavedLicenseKey(
   System.Type type,
   System.String key
);
```

LicenseException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LicenseException
   extends System.SystemException
```

Public Constructors

```
public LicenseException(
   System.Type type
);
public LicenseException(
   System.Type type,
   System.Object instance
);
public LicenseException(
   System.Type type,
   System.Object instance,
   System.String message
);
public LicenseException(
   System.Type type,
   System.Object instance,
   System.String message,
   System.Exception innerException
);
```

Public Properties

```
public System.Type get_LicensedType();
```

LicenseManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LicenseManager
   extends System.Object
```

Public Properties

```
public static System.ComponentModel.LicenseContext get_CurrentContext();
public static void set_CurrentContext(
   System.ComponentModel.LicenseContext value
);
public static System.ComponentModel.LicenseUsageMode get_UsageMode();
```

Public Methods

```
public static System.Object CreateWithContext(
   System.Type type,
   System.ComponentModel.LicenseContext creationContext
);
public static System.Object CreateWithContext(
   System.Type type,
   System.ComponentModel.LicenseContext creationContext,
   System.Object[] args
);
public static boolean IsLicensed(
   System.Type type
);
public static boolean IsValid(
   System.Type type
);
public static boolean IsValid(
   System.Type type,
   System.Object instance,
   System.ComponentModel.License& license
);
public static void LockContext(
   System.Object contextUser
);
public static void UnlockContext(
   System.Object contextUser
);
public static void Validate(
   System.Type type
);
public static System.ComponentModel.License Validate(
   System.Type type,
   System.Object instance
);
```

LicenseProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class LicenseProvider
   extends System.Object
```

Public Methods

```
public abstract System.ComponentModel.License GetLicense(
   System.ComponentModel.LicenseContext context,
   System.Type type,
   System.Object instance,
   boolean allowExceptions
);
```

Protected Constructors

```
protected LicenseProvider();
```

LicenseProviderAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = false) */
public class LicenseProviderAttribute
   extends System.Attribute
```

Public Constructors

```
public LicenseProviderAttribute();
public LicenseProviderAttribute(
   System.String typeName
);
public LicenseProviderAttribute(
   System.Type type
);
```

Public Properties

```
public System.Type get_LicenseProvider();
public System.Object get_TypeId();
```

Public Methods

```
public boolean Equals(
   System.Object value
);
public int GetHashCode();
```

Public Fields

```
public static final System.ComponentModel.LicenseProviderAttribute Default;
```

LicenseUsageMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum LicenseUsageMode
```

Members

Designtime

Runtime

LicFileLicenseProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LicFileLicenseProvider
   extends System.ComponentModel.LicenseProvider
```

Public Constructors

```
public LicFileLicenseProvider();
```

Public Methods

```
public System.ComponentModel.License GetLicense(
   System.ComponentModel.LicenseContext context,
   System.Type type,
   System.Object instance,
   boolean allowExceptions
);
```

Protected Methods

```
protected System.String GetKey(
   System.Type type
);
protected boolean IsKeyValid(
   System.String key,
   System.Type type
);
```

ListBindableAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class ListBindableAttribute
   extends System.Attribute
```

Public Constructors

```
public ListBindableAttribute(
   boolean listBindable
);
public ListBindableAttribute(
   System.ComponentModel.BindableSupport flags
);
```

Public Properties

```
public boolean get_ListBindable();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.ListBindableAttribute Default;
public static final System.ComponentModel.ListBindableAttribute No;
public static final System.ComponentModel.ListBindableAttribute Yes;
```

ListChangedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ListChangedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ListChangedEventArgs(
   System.ComponentModel.ListChangedType listChangedType,
   int newIndex
);
public ListChangedEventArgs(
   System.ComponentModel.ListChangedType listChangedType,
   System.ComponentModel.PropertyDescriptor propDesc
);
public ListChangedEventArgs(
   System.ComponentModel.ListChangedType listChangedType,
   int newIndex,
   int oldIndex
);
```

Public Properties

```
public System.ComponentModel.ListChangedType get_ListChangedType();
public int get_NewIndex();
public int get_OldIndex();
```

ListChangedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ListChangedEventHandler(
   System.Object sender,
   System.ComponentModel.ListChangedEventArgs e
);
```

ListChangedType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ListChangedType
```

Members

ItemAdded

ItemChanged

ItemDeleted

ItemMoved

PropertyDescriptorAdded

PropertyDescriptorChanged

PropertyDescriptorDeleted

Reset

ListSortDirection Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ListSortDirection
```

Members

Ascending

Descending

LocalizableAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class LocalizableAttribute
   extends System.Attribute
```

Public Constructors

```
public LocalizableAttribute(
   boolean isLocalizable
);
```

Public Properties

```
public boolean get_IsLocalizable();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.LocalizableAttribute Default;
public static final System.ComponentModel.LocalizableAttribute No;
public static final System.ComponentModel.LocalizableAttribute Yes;
```

MarshalByValueComponent Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IRootDesigner, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77
a5c561934e089", DesignerTypeName = "System.Windows.Forms.Design.ComponentDocumentDesigner, Sy
stem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DesignerCategoryAttribute(Category = "Component") */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Compo
nentModel.ComponentConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77
a5c561934e089") */
public class MarshalByValueComponent
   extends System.Object
   implements System.ComponentModel.IComponent, System.IDisposable, System.IServiceProvider
```

Public Constructors

```
public MarshalByValueComponent();
```

Public Properties

```
public System.ComponentModel.IContainer get_Container();
public boolean get_DesignMode();
public System.ComponentModel.ISite get_Site();
public void set_Site(
   System.ComponentModel.ISite value
);
```

Public Methods

```
public final void Dispose();
public System.Object GetService(
   System.Type service
);
public System.String ToString();
```

Public Events

```
public final void add_Disposed(
   System.EventHandler value
);
public final void remove_Disposed(
   System.EventHandler value
);
```

Protected Properties

```
protected System.ComponentModel.EventHandlerList get_Events();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

MemberDescriptor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public abstract class MemberDescriptor
   extends System.Object
```

Public Properties

```
public System.ComponentModel.AttributeCollection get_Attributes();
public System.String get_Category();
public System.String get_Description();
public boolean get_DesignTimeOnly();
public System.String get_DisplayName();
public boolean get_IsBrowsable();
public System.String get_Name();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

Protected Constructors

```
protected MemberDescriptor(
   System.ComponentModel.MemberDescriptor descr
);
protected MemberDescriptor(
   System.String name
);
protected MemberDescriptor(
   System.ComponentModel.MemberDescriptor oldMemberDescriptor,
   System.Attribute[] newAttributes
);
protected MemberDescriptor(
   System.String name,
   System.Attribute[] attributes
);
```

Protected Properties

```
protected System.Attribute[] get_AttributeArray();
protected void set_AttributeArray(
   System.Attribute[] value
);
protected int get_NameHashCode();
```

Protected Methods

```
protected System.ComponentModel.AttributeCollection CreateAttributeCollection();
```

```
protected void FillAttributes(
   System.Collections.IList attributeList
);
protected static System.Reflection.MethodInfo FindMethod(
   System.Type componentClass,
   System.String name,
   System.Type[] args,
   System.Type returnType
);
protected static System.Reflection.MethodInfo FindMethod(
   System.Type componentClass,
   System.String name,
   System.Type[] args,
   System.Type returnType,
   boolean publicOnly
);
protected static System.Object GetInvokee(
   System.Type componentClass,
   System.Object component
);
protected static System.ComponentModel.ISite GetSite(
   System.Object component
);
```

MergablePropertyAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class MergablePropertyAttribute
   extends System.Attribute
```

Public Constructors

```
public MergablePropertyAttribute(
   boolean allowMerge
);
```

Public Properties

```
public boolean get_AllowMerge();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.MergablePropertyAttribute Default;
public static final System.ComponentModel.MergablePropertyAttribute No;
public static final System.ComponentModel.MergablePropertyAttribute Yes;
```

NotifyParentPropertyAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, AllowMultiple = false, Inhe
rited = true) */
public class NotifyParentPropertyAttribute
   extends System.Attribute
```

Public Constructors

```
public NotifyParentPropertyAttribute(
   boolean notifyParent
);
```

Public Properties

```
public boolean get_NotifyParent();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.NotifyParentPropertyAttribute Default;
public static final System.ComponentModel.NotifyParentPropertyAttribute No;
public static final System.ComponentModel.NotifyParentPropertyAttribute Yes;
```

ParenthesizePropertyNameAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class ParenthesizePropertyNameAttribute
   extends System.Attribute
```

Public Constructors

```
public ParenthesizePropertyNameAttribute();
public ParenthesizePropertyNameAttribute(
   boolean needParenthesis
);
```

Public Properties

```
public boolean get_NeedParenthesis();
```

Public Methods

```
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.ParenthesizePropertyNameAttribute Default;
```

PropertyChangedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PropertyChangedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public PropertyChangedEventArgs(
   System.String propertyName
);
```

Public Properties

```
public System.String get_PropertyName();
```

PropertyChangedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void PropertyChangedEventHandler(
   System.Object sender,
   System.ComponentModel.PropertyChangedEventArgs e
);
```

PropertyDescriptor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public abstract class PropertyDescriptor
   extends System.ComponentModel.MemberDescriptor
```

Public Properties

```
public abstract System.Type get_ComponentType();
public System.ComponentModel.TypeConverter get_Converter();
public boolean get_IsLocalizable();
public abstract boolean get_IsReadOnly();
public abstract System.Type get_PropertyType();
public System.ComponentModel.DesignerSerializationVisibility get_SerializationVisibility();
```

Public Methods

```
public void AddValueChanged(
   System.Object component,
   System.EventHandler handler
);
public abstract boolean CanResetValue(
   System.Object component
);
public boolean Equals(
   System.Object obj
);
public System.ComponentModel.PropertyDescriptorCollection GetChildProperties();
public System.ComponentModel.PropertyDescriptorCollection GetChildProperties(
   System.Attribute[] filter
);
public System.ComponentModel.PropertyDescriptorCollection GetChildProperties(
   System.Object instance
);
public System.ComponentModel.PropertyDescriptorCollection GetChildProperties(
   System.Object instance,
   System.Attribute[] filter
);
public System.Object GetEditor(
   System.Type editorBaseType
);
public int GetHashCode();
public abstract System.Object GetValue(
   System.Object component
);
public void RemoveValueChanged(
   System.Object component,
   System.EventHandler handler
);
public abstract void ResetValue(
   System.Object component
);
public abstract void SetValue(
   System.Object component,
   System.Object value
);
public abstract boolean ShouldSerializeValue(
```

```
   System.Object component
);
```

Protected Constructors

```
protected PropertyDescriptor(
   System.ComponentModel.MemberDescriptor descr
);
protected PropertyDescriptor(
   System.ComponentModel.MemberDescriptor descr,
   System.Attribute[] attrs
);
protected PropertyDescriptor(
   System.String name,
   System.Attribute[] attrs
);
```

Protected Methods

```
protected System.Object CreateInstance(
   System.Type type
);
protected System.Type GetTypeFromName(
   System.String typeName
);
protected void OnValueChanged(
   System.Object component,
   System.EventArgs e
);
```

PropertyDescriptorCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class PropertyDescriptorCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable, System.Collections.IDictionary
```

Public Constructors

```
public PropertyDescriptorCollection(
   System.ComponentModel.PropertyDescriptor[] properties
);
```

Public Properties

```
public int get_Count();
public System.ComponentModel.PropertyDescriptor get_Item(
   int index
);
public System.ComponentModel.PropertyDescriptor get_Item(
   System.String name
);
```

Public Methods

```
public int Add(
   System.ComponentModel.PropertyDescriptor value
);
public void Clear();
public boolean Contains(
   System.ComponentModel.PropertyDescriptor value
);
public final void CopyTo(
   System.Array array,
   int index
);
public System.ComponentModel.PropertyDescriptor Find(
   System.String name,
   boolean ignoreCase
);
public System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.ComponentModel.PropertyDescriptor value
);
public void Insert(
   int index,
   System.ComponentModel.PropertyDescriptor value
);
public void Remove(
   System.ComponentModel.PropertyDescriptor value
);
public void RemoveAt(
   int index
);
```

```
public System.ComponentModel.PropertyDescriptorCollection Sort();
public System.ComponentModel.PropertyDescriptorCollection Sort(
   System.Collections.IComparer comparer
);
public System.ComponentModel.PropertyDescriptorCollection Sort(
   System.String[] names
);
public System.ComponentModel.PropertyDescriptorCollection Sort(
   System.String[] names,
   System.Collections.IComparer comparer
);
```

Public Fields

```
public static final System.ComponentModel.PropertyDescriptorCollection Empty;
```

Protected Methods

```
protected void InternalSort(
   System.Collections.IComparer sorter
);
protected void InternalSort(
   System.String[] names
);
```

PropertyTabAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class PropertyTabAttribute
   extends System.Attribute
```

Public Constructors

```
public PropertyTabAttribute();
public PropertyTabAttribute(
   System.String tabClassName
);
public PropertyTabAttribute(
   System.Type tabClass
);
public PropertyTabAttribute(
   System.String tabClassName,
   System.ComponentModel.PropertyTabScope tabScope
);
public PropertyTabAttribute(
   System.Type tabClass,
   System.ComponentModel.PropertyTabScope tabScope
);
```

Public Properties

```
public System.Type[] get_TabClasses();
public System.ComponentModel.PropertyTabScope[] get_TabScopes();
```

Public Methods

```
public boolean Equals(
   System.ComponentModel.PropertyTabAttribute other
);
public boolean Equals(
   System.Object other
);
public int GetHashCode();
```

Protected Properties

```
protected System.String[] get_TabClassNames();
```

Protected Methods

```
protected void InitializeArrays(
   System.String[] tabClassNames,
   System.ComponentModel.PropertyTabScope[] tabScopes
);
protected void InitializeArrays(
```

```
    System.Type[] tabClasses,
    System.ComponentModel.PropertyTabScope[] tabScopes
);
```

PropertyTabScope Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PropertyTabScope
```

Members

Component

Document

Global

Static

ProvidePropertyAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = true, Inherite
d = true) */
public class ProvidePropertyAttribute
   extends System.Attribute
```

Public Constructors

```
public ProvidePropertyAttribute(
   System.String propertyName,
   System.String receiverTypeName
);
public ProvidePropertyAttribute(
   System.String propertyName,
   System.Type receiverType
);
```

Public Properties

```
public System.String get_PropertyName();
public System.String get_ReceiverTypeName();
public System.Object get_TypeId();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

ReadOnlyAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class ReadOnlyAttribute
   extends System.Attribute
```

Public Constructors

```
public ReadOnlyAttribute(
   boolean isReadOnly
);
```

Public Properties

```
public boolean get_IsReadOnly();
```

Public Methods

```
public boolean Equals(
   System.Object value
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.ReadOnlyAttribute Default;
public static final System.ComponentModel.ReadOnlyAttribute No;
public static final System.ComponentModel.ReadOnlyAttribute Yes;
```

RecommendedAsConfigurableAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, AllowMultiple = false, Inhe
rited = true) */
public class RecommendedAsConfigurableAttribute
   extends System.Attribute
```

Public Constructors

```
public RecommendedAsConfigurableAttribute(
   boolean recommendedAsConfigurable
);
```

Public Properties

```
public boolean get_RecommendedAsConfigurable();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.RecommendedAsConfigurableAttribute Default;
public static final System.ComponentModel.RecommendedAsConfigurableAttribute No;
public static final System.ComponentModel.RecommendedAsConfigurableAttribute Yes;
```

ReferenceConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ReferenceConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public ReferenceConverter(
   System.Type type
);
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

Protected Methods

```
protected boolean IsValueAllowed(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value
);
```

RefreshEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RefreshEventArgs
   extends System.EventArgs
```

Public Constructors

```
public RefreshEventArgs(
   System.Object componentChanged
);
public RefreshEventArgs(
   System.Type typeChanged
);
```

Public Properties

```
public System.Object get_ComponentChanged();
public System.Type get_TypeChanged();
```

RefreshEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void RefreshEventHandler(
   System.ComponentModel.RefreshEventArgs e
);
```

RefreshProperties Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum RefreshProperties
```

Members

All

None

Repaint

RefreshPropertiesAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class RefreshPropertiesAttribute
   extends System.Attribute
```

Public Constructors

```
public RefreshPropertiesAttribute(
   System.ComponentModel.RefreshProperties refresh
);
```

Public Properties

```
public System.ComponentModel.RefreshProperties get_RefreshProperties();
```

Public Methods

```
public boolean Equals(
   System.Object value
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.RefreshPropertiesAttribute All;
public static final System.ComponentModel.RefreshPropertiesAttribute Default;
public static final System.ComponentModel.RefreshPropertiesAttribute Repaint;
```

RunInstallerAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class RunInstallerAttribute
   extends System.Attribute
```

Public Constructors

```
public RunInstallerAttribute(
   boolean runInstaller
);
```

Public Properties

```
public boolean get_RunInstaller();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.RunInstallerAttribute Default;
public static final System.ComponentModel.RunInstallerAttribute No;
public static final System.ComponentModel.RunInstallerAttribute Yes;
```

SByteConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SByteConverter
   extends System.ComponentModel.BaseNumberConverter
```

Public Constructors

```
public SByteConverter();
```

SingleConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SingleConverter
   extends System.ComponentModel.BaseNumberConverter
```

Public Constructors

```
public SingleConverter();
```

StringConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class StringConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public StringConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
```

SyntaxCheck Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SyntaxCheck
   extends System.Object
```

Public Methods

```
public static boolean CheckMachineName(
   System.String value
);
public static boolean CheckPath(
   System.String value
);
public static boolean CheckRootedPath(
   System.String value
);
```

TimeSpanConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TimeSpanConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public TimeSpanConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

ToolboxItemAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class ToolboxItemAttribute
   extends System.Attribute
```

Public Constructors

```
public ToolboxItemAttribute(
   boolean defaultType
);
public ToolboxItemAttribute(
   System.String toolboxItemTypeName
);
public ToolboxItemAttribute(
   System.Type toolboxItemType
);
```

Public Properties

```
public System.Type get_ToolboxItemType();
public System.String get_ToolboxItemTypeName();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.ComponentModel.ToolboxItemAttribute Default;
public static final System.ComponentModel.ToolboxItemAttribute None;
```

ToolboxItemFilterAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = true, Inherite
d = true) */
public class ToolboxItemFilterAttribute
   extends System.Attribute
```

Public Constructors

```
public ToolboxItemFilterAttribute(
   System.String filterString
);
public ToolboxItemFilterAttribute(
   System.String filterString,
   System.ComponentModel.ToolboxItemFilterType filterType
);
```

Public Properties

```
public System.String get_FilterString();
public System.ComponentModel.ToolboxItemFilterType get_FilterType();
public System.Object get_TypeId();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean Match(
   System.Object obj
);
```

ToolboxItemFilterType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ToolboxItemFilterType
```

Members

Allow

Custom

Prevent

Require

TypeConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class TypeConverter
   extends System.Object
```

Public Constructors

```
public TypeConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.Type sourceType
);
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.Type destinationType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.Object value
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertFromInvariantString(
   System.String text
);
public System.Object ConvertFromInvariantString(
   System.ComponentModel.ITypeDescriptorContext context,
   System.String text
);
public System.Object ConvertFromString(
   System.String text
);
public System.Object ConvertFromString(
   System.ComponentModel.ITypeDescriptorContext context,
   System.String text
);
public System.Object ConvertFromString(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.String text
);
public System.Object ConvertTo(
   System.Object value,
   System.Type destinationType
);
```

```
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.String ConvertToInvariantString(
   System.Object value
);
public System.String ConvertToInvariantString(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value
);
public System.String ConvertToString(
   System.Object value
);
public System.String ConvertToString(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value
);
public System.String ConvertToString(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object CreateInstance(
   System.Collections.IDictionary propertyValues
);
public System.Object CreateInstance(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Collections.IDictionary propertyValues
);
public boolean GetCreateInstanceSupported();
public boolean GetCreateInstanceSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.Object value
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value,
   System.Attribute[] attributes
);
public boolean GetPropertiesSupported();
public boolean GetPropertiesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public System.Collections.ICollection GetStandardValues();
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive();
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported();
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean IsValid(
   System.Object value
);
public boolean IsValid(
```

```
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value
);
```

Protected Methods

```
protected System.Exception GetConvertFromException(
   System.Object value
);
protected System.Exception GetConvertToException(
   System.Object value,
   System.Type destinationType
);
protected System.ComponentModel.PropertyDescriptorCollection SortProperties(
   System.ComponentModel.PropertyDescriptorCollection props,
   System.String[] names
);
```

TypeConverter.SimplePropertyDescriptor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
protected abstract class SimplePropertyDescriptor
   extends System.ComponentModel.PropertyDescriptor
```

Public Constructors

```
public SimplePropertyDescriptor(
   System.Type componentType,
   System.String name,
   System.Type propertyType
);
public SimplePropertyDescriptor(
   System.Type componentType,
   System.String name,
   System.Type propertyType,
   System.Attribute[] attributes
);
```

Public Properties

```
public System.Type get_ComponentType();
public boolean get_IsReadOnly();
public System.Type get_PropertyType();
```

Public Methods

```
public boolean CanResetValue(
   System.Object component
);
public void ResetValue(
   System.Object component
);
public boolean ShouldSerializeValue(
   System.Object component
);
```

TypeConverter.StandardValuesCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class StandardValuesCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public StandardValuesCollection(
   System.Collections.ICollection values
);
```

Public Properties

```
public int get_Count();
public System.Object get_Item(
   int index
);
```

Public Methods

```
public void CopyTo(
   System.Array array,
   int index
);
public System.Collections.IEnumerator GetEnumerator();
```

TypeConverterAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class TypeConverterAttribute
   extends System.Attribute
```

Public Constructors

```
public TypeConverterAttribute();
public TypeConverterAttribute(
   System.String typeName
);
public TypeConverterAttribute(
   System.Type type
);
```

Public Properties

```
public System.String get_ConverterTypeName();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

Public Fields

```
public static final System.ComponentModel.TypeConverterAttribute Default;
```

TypeDescriptor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TypeDescriptor
   extends System.Object
```

Public Properties

```
public static System.ComponentModel.IComNativeDescriptorHandler get_ComNativeDescriptorHandle
r();
public static void set_ComNativeDescriptorHandler(
   System.ComponentModel.IComNativeDescriptorHandler value
);
```

Public Methods

```
public static void AddEditorTable(
   System.Type editorBaseType,
   System.Collections.Hashtable table
);
public static System.ComponentModel.Design.IDesigner CreateDesigner(
   System.ComponentModel.IComponent component,
   System.Type designerBaseType
);
public static System.ComponentModel.EventDescriptor CreateEvent(
   System.Type componentType,
   System.ComponentModel.EventDescriptor oldEventDescriptor,
   System.Attribute[] attributes
);
public static System.ComponentModel.EventDescriptor CreateEvent(
   System.Type componentType,
   System.String name,
   System.Type type,
   System.Attribute[] attributes
);
public static System.ComponentModel.PropertyDescriptor CreateProperty(
   System.Type componentType,
   System.ComponentModel.PropertyDescriptor oldPropertyDescriptor,
   System.Attribute[] attributes
);
public static System.ComponentModel.PropertyDescriptor CreateProperty(
   System.Type componentType,
   System.String name,
   System.Type type,
   System.Attribute[] attributes
);
public static System.ComponentModel.AttributeCollection GetAttributes(
   System.Object component
);
public static System.ComponentModel.AttributeCollection GetAttributes(
   System.Type componentType
);
public static System.ComponentModel.AttributeCollection GetAttributes(
   System.Object component,
   boolean noCustomTypeDesc
);
public static System.String GetClassName(
   System.Object component
```

```
);
public static System.String GetClassName(
   System.Object component,
   boolean noCustomTypeDesc
);
public static System.String GetComponentName(
   System.Object component
);
public static System.String GetComponentName(
   System.Object component,
   boolean noCustomTypeDesc
);
public static System.ComponentModel.TypeConverter GetConverter(
   System.Object component
);
public static System.ComponentModel.TypeConverter GetConverter(
   System.Type type
);
public static System.ComponentModel.TypeConverter GetConverter(
   System.Object component,
   boolean noCustomTypeDesc
);
public static System.ComponentModel.EventDescriptor GetDefaultEvent(
   System.Object component
);
public static System.ComponentModel.EventDescriptor GetDefaultEvent(
   System.Type componentType
);
public static System.ComponentModel.EventDescriptor GetDefaultEvent(
   System.Object component,
   boolean noCustomTypeDesc
);
public static System.ComponentModel.PropertyDescriptor GetDefaultProperty(
   System.Object component
);
public static System.ComponentModel.PropertyDescriptor GetDefaultProperty(
   System.Type componentType
);
public static System.ComponentModel.PropertyDescriptor GetDefaultProperty(
   System.Object component,
   boolean noCustomTypeDesc
);
public static System.Object GetEditor(
   System.Object component,
   System.Type editorBaseType
);
public static System.Object GetEditor(
   System.Type type,
   System.Type editorBaseType
);
public static System.Object GetEditor(
   System.Object component,
   System.Type editorBaseType,
   boolean noCustomTypeDesc
);
public static System.ComponentModel.EventDescriptorCollection GetEvents(
   System.Object component
);
public static System.ComponentModel.EventDescriptorCollection GetEvents(
   System.Type componentType
);
public static System.ComponentModel.EventDescriptorCollection GetEvents(
   System.Object component,
   boolean noCustomTypeDesc
);
public static System.ComponentModel.EventDescriptorCollection GetEvents(
   System.Object component,
   System.Attribute[] attributes
```

```
);
public static System.ComponentModel.EventDescriptorCollection GetEvents(
   System.Type componentType,
   System.Attribute[] attributes
);
public static System.ComponentModel.EventDescriptorCollection GetEvents(
   System.Object component,
   System.Attribute[] attributes,
   boolean noCustomTypeDesc
);
public static System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.Object component
);
public static System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.Type componentType
);
public static System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.Object component,
   boolean noCustomTypeDesc
);
public static System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.Object component,
   System.Attribute[] attributes
);
public static System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.Type componentType,
   System.Attribute[] attributes
);
public static System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.Object component,
   System.Attribute[] attributes,
   boolean noCustomTypeDesc
);
public static void Refresh(
   System.Object component
);
public static void Refresh(
   System.Reflection.Assembly assembly
);
public static void Refresh(
   System.Reflection.Module module
);
public static void Refresh(
   System.Type type
);
public static void SortDescriptorArray(
   System.Collections.IList infos
);
```

Public Events

```
public static void add_Refreshed(
   System.ComponentModel.RefreshEventHandler value
);
public static void remove_Refreshed(
   System.ComponentModel.RefreshEventHandler value
);
```

TypeListConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class TypeListConverter
   extends System.ComponentModel.TypeConverter
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

Protected Constructors

```
protected TypeListConverter(
   System.Type[] types
);
```

UInt16Converter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UInt16Converter
   extends System.ComponentModel.BaseNumberConverter
```

Public Constructors

```
public UInt16Converter();
```

UInt32Converter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UInt32Converter
   extends System.ComponentModel.BaseNumberConverter
```

Public Constructors

```
public UInt32Converter();
```

UInt64Converter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UInt64Converter
   extends System.ComponentModel.BaseNumberConverter
```

Public Constructors

```
public UInt64Converter();
```

WarningException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WarningException
   extends System.SystemException
```

Public Constructors

```
public WarningException(
   System.String message
);
public WarningException(
   System.String message,
   System.String helpUrl
);
public WarningException(
   System.String message,
   System.String helpUrl,
   System.String helpTopic
);
```

Public Properties

```
public System.String get_HelpTopic();
public System.String get_HelpUrl();
```

Win32Exception Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Security.SuppressUnmanagedCodeSecurityAttribute() */
public class Win32Exception
   extends System.Runtime.InteropServices.ExternalException
```

Public Constructors

```
public Win32Exception();
public Win32Exception(
   int error
);
public Win32Exception(
   int error,
   System.String message
);
```

Public Properties

```
public int get_NativeErrorCode();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected Win32Exception(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

System.ComponentModel.Design Namespace (Visual J# Syntax)

Classes

ActiveDesignerEventArgs

ArrayEditor

BinaryEditor

ByteViewer

CheckoutException

CollectionEditor

CollectionEditor.CollectionForm

CommandID

ComponentChangedEventArgs

ComponentChangingEventArgs

ComponentDesigner

ComponentDesigner.ShadowPropertyCollection

ComponentEventArgs

ComponentRenameEventArgs

DateTimeEditor

DesignerCollection

DesignerEventArgs

DesignerTransaction

DesignerTransactionCloseEventArgs

DesignerVerb

DesignerVerbCollection

DesigntimeLicenseContext

DesigntimeLicenseContextSerializer

InheritanceService

LocalizationExtenderProvider

MenuCommand

ObjectSelectorEditor

ObjectSelectorEditor.Selector

ObjectSelectorEditor.SelectorNode

ServiceContainer

StandardCommands

StandardToolWindows

Interfaces

IComponentChangeService

IDesigner

IDesignerEventService

IDesignerFilter

IDesignerHost

IDesignerOptionService

IDictionaryService

IEventBindingService

IExtenderListService

IExtenderProviderService

IHelpService

IInheritanceService

IMenuCommandService

IReferenceService

IResourceService

IRootDesigner

ISelectionService

IServiceContainer

ITypeDescriptorFilterService

ITypeResolutionService

Delegates

ActiveDesignerEventHandler

ComponentChangedEventHandler

ComponentChangingEventHandler

ComponentEventHandler

ComponentRenameEventHandler

DesignerEventHandler

DesignerTransactionCloseEventHandler

ServiceCreatorCallback

Enumerations

DisplayMode

HelpContextType

HelpKeywordType

SelectionTypes

ViewTechnology

ActiveDesignerEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ActiveDesignerEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ActiveDesignerEventArgs(
   System.ComponentModel.Design.IDesignerHost oldDesigner,
   System.ComponentModel.Design.IDesignerHost newDesigner
);
```

Public Properties

```
public System.ComponentModel.Design.IDesignerHost get_NewDesigner();
public System.ComponentModel.Design.IDesignerHost get_OldDesigner();
```

ActiveDesignerEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ActiveDesignerEventHandler(
   System.Object sender,
   System.ComponentModel.Design.ActiveDesignerEventArgs e
);
```

ArrayEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ArrayEditor
   extends System.ComponentModel.Design.CollectionEditor
```

Public Constructors

```
public ArrayEditor(
   System.Type type
);
```

Protected Methods

```
protected System.Type CreateCollectionItemType();
protected System.Object[] GetItems(
   System.Object editValue
);
protected System.Object SetItems(
   System.Object editValue,
   System.Object[] value
);
```

BinaryEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BinaryEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public BinaryEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

ByteViewer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
public class ByteViewer
   extends System.Windows.Forms.Control
```

Public Constructors

```
public ByteViewer();
```

Public Methods

```
public ubyte[] GetBytes();
public System.ComponentModel.Design.DisplayMode GetDisplayMode();
public void SaveToFile(
   System.String path
);
public void SetBytes(
   ubyte[] bytes
);
public void SetDisplayMode(
   System.ComponentModel.Design.DisplayMode mode
);
public void SetFile(
   System.String path
);
public void SetStartLine(
   int line
);
```

Protected Methods

```
protected void OnKeyDown(
   System.Windows.Forms.KeyEventArgs e
);
protected void OnPaint(
   System.Windows.Forms.PaintEventArgs e
);
protected void OnResize(
   System.EventArgs e
);
protected void ScrollChanged(
   System.Object source,
   System.EventArgs e
);
```

CheckoutException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CheckoutException
   extends System.Runtime.InteropServices.ExternalException
```

Public Constructors

```
public CheckoutException();
public CheckoutException(
   System.String message
);
public CheckoutException(
   System.String message,
   int errorCode
);
```

Public Fields

```
public static final System.ComponentModel.Design.CheckoutException Canceled;
```

CollectionEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CollectionEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public CollectionEditor(
   System.Type type
);
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

Protected Properties

```
protected System.Type get_CollectionItemType();
protected System.Type get_CollectionType();
protected System.ComponentModel.ITypeDescriptorContext get_Context();
protected System.String get_HelpTopic();
protected System.Type[] get_NewItemTypes();
```

Protected Methods

```
protected boolean CanRemoveInstance(
   System.Object value
);
protected boolean CanSelectMultipleInstances();
protected System.ComponentModel.Design.CollectionEditor+CollectionForm CreateCollectionForm()
;
protected System.Type CreateCollectionItemType();
protected System.Object CreateInstance(
   System.Type itemType
);
protected System.Type[] CreateNewItemTypes();
protected void DestroyInstance(
   System.Object instance
);
protected System.Object[] GetItems(
   System.Object editValue
);
protected System.Object GetService(
   System.Type serviceType
);
protected System.Object SetItems(
```

```
    System.Object editValue,
    System.Object[] value
);
protected void ShowHelp();
```

CollectionEditor.CollectionForm Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
protected abstract class CollectionForm
   extends System.Windows.Forms.Form
```

Public Constructors

```
public CollectionForm(
   System.ComponentModel.Design.CollectionEditor editor
);
```

Public Properties

```
public System.Object get_EditValue();
public void set_EditValue(
   System.Object value
);
```

Protected Properties

```
protected System.Type get_CollectionItemType();
protected System.Type get_CollectionType();
protected System.ComponentModel.ITypeDescriptorContext get_Context();
protected System.Object[] get_Items();
protected void set_Items(
   System.Object[] value
);
protected System.Type[] get_NewItemTypes();
```

Protected Methods

```
protected boolean CanRemoveInstance(
   System.Object value
);
protected boolean CanSelectMultipleInstances();
protected System.Object CreateInstance(
   System.Type itemType
);
protected void DestroyInstance(
   System.Object instance
);
protected void DisplayError(
   System.Exception e
);
protected System.Object GetService(
   System.Type serviceType
);
protected abstract void OnEditValueChanged();
protected System.Windows.Forms.DialogResult ShowEditorDialog(
   System.Windows.Forms.Design.IWindowsFormsEditorService edSvc
);
```

CommandID Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CommandID
   extends System.Object
```

Public Constructors

```
public CommandID(
   System.Guid menuGroup,
   int commandID
);
```

Public Properties

```
public System.Guid get_Guid();
public int get_ID();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public System.String ToString();
```

ComponentChangedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class ComponentChangedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ComponentChangedEventArgs(
   System.Object component,
   System.ComponentModel.MemberDescriptor member,
   System.Object oldValue,
   System.Object newValue
);
```

Public Properties

```
public System.Object get_Component();
public System.ComponentModel.MemberDescriptor get_Member();
public System.Object get_NewValue();
public System.Object get_OldValue();
```

ComponentChangedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ComponentChangedEventHandler(
   System.Object sender,
   System.ComponentModel.Design.ComponentChangedEventArgs e
);
```

ComponentChangingEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class ComponentChangingEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ComponentChangingEventArgs(
   System.Object component,
   System.ComponentModel.MemberDescriptor member
);
```

Public Properties

```
public System.Object get_Component();
public System.ComponentModel.MemberDescriptor get_Member();
```

ComponentChangingEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ComponentChangingEventHandler(
   System.Object sender,
   System.ComponentModel.Design.ComponentChangingEventArgs e
);
```

ComponentDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ComponentDesigner
   extends System.Object
   implements System.ComponentModel.Design.IDesigner, System.IDisposable, System.ComponentMod
el.Design.IDesignerFilter
```

Public Constructors

```
public ComponentDesigner();
```

Public Properties

```
public System.Collections.ICollection get_AssociatedComponents();
public final System.ComponentModel.IComponent get_Component();
public System.ComponentModel.Design.DesignerVerbCollection get_Verbs();
```

Public Methods

```
public final void Dispose();
public void DoDefaultAction();
public void Initialize(
   System.ComponentModel.IComponent component
);
public void InitializeNonDefault();
public void OnSetComponentDefaults();
```

Protected Properties

```
protected System.ComponentModel.InheritanceAttribute get_InheritanceAttribute();
protected boolean get_Inherited();
protected System.ComponentModel.Design.ComponentDesigner+ShadowPropertyCollection get_ShadowP
roperties();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
protected System.Object GetService(
   System.Type serviceType
);
protected System.ComponentModel.InheritanceAttribute InvokeGetInheritanceAttribute(
   System.ComponentModel.Design.ComponentDesigner toInvoke
);
protected void PostFilterAttributes(
   System.Collections.IDictionary attributes
);
protected void PostFilterEvents(
   System.Collections.IDictionary events
```

```
);
protected void PostFilterProperties(
   System.Collections.IDictionary properties
);
protected void PreFilterAttributes(
   System.Collections.IDictionary attributes
);
protected void PreFilterEvents(
   System.Collections.IDictionary events
);
protected void PreFilterProperties(
   System.Collections.IDictionary properties
);
protected void RaiseComponentChanged(
   System.ComponentModel.MemberDescriptor member,
   System.Object oldValue,
   System.Object newValue
);
protected void RaiseComponentChanging(
   System.ComponentModel.MemberDescriptor member
);
```

ComponentDesigner.ShadowPropertyCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
protected class ShadowPropertyCollection
   extends System.Object
```

Public Properties

```
public System.Object get_Item(
   System.String propertyName
);
public void set_Item(
   System.String propertyName,
   System.Object value
);
```

Public Methods

```
public boolean Contains(
   System.String propertyName
);
```

ComponentEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class ComponentEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ComponentEventArgs(
   System.ComponentModel.IComponent component
);
```

Public Properties

```
public System.ComponentModel.IComponent get_Component();
```

ComponentEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ComponentEventHandler(
   System.Object sender,
   System.ComponentModel.Design.ComponentEventArgs e
);
```

ComponentRenameEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class ComponentRenameEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ComponentRenameEventArgs(
   System.Object component,
   System.String oldName,
   System.String newName
);
```

Public Properties

```
public System.Object get_Component();
public System.String get_NewName();
public System.String get_OldName();
```

ComponentRenameEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ComponentRenameEventHandler(
   System.Object sender,
   System.ComponentModel.Design.ComponentRenameEventArgs e
);
```

DateTimeEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DateTimeEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public DateTimeEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

DesignerCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DesignerCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public DesignerCollection(
   System.Collections.IList designers
);
public DesignerCollection(
   System.ComponentModel.Design.IDesignerHost[] designers
);
```

Public Properties

```
public int get_Count();
public System.ComponentModel.Design.IDesignerHost get_Item(
   int index
);
```

Public Methods

```
public System.Collections.IEnumerator GetEnumerator();
```

DesignerEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DesignerEventArgs
   extends System.EventArgs
```

Public Constructors

```
public DesignerEventArgs(
   System.ComponentModel.Design.IDesignerHost host
);
```

Public Properties

```
public System.ComponentModel.Design.IDesignerHost get_Designer();
```

DesignerEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DesignerEventHandler(
   System.Object sender,
   System.ComponentModel.Design.DesignerEventArgs e
);
```

DesignerTransaction Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class DesignerTransaction
   extends System.Object
   implements System.IDisposable
```

Public Constructors

```
public DesignerTransaction();
public DesignerTransaction(
   System.String description
);
```

Public Properties

```
public boolean get_Canceled();
public boolean get_Committed();
public System.String get_Description();
```

Public Methods

```
public void Cancel();
public void Commit();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
protected abstract void OnCancel();
protected abstract void OnCommit();
```

DesignerTransactionCloseEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class DesignerTransactionCloseEventArgs
   extends System.EventArgs
```

Public Constructors

```
public DesignerTransactionCloseEventArgs(
   boolean commit
);
```

Public Properties

```
public boolean get_TransactionCommitted();
```

DesignerTransactionCloseEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DesignerTransactionCloseEventHandler(
   System.Object sender,
   System.ComponentModel.Design.DesignerTransactionCloseEventArgs e
);
```

DesignerVerb Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class DesignerVerb
   extends System.ComponentModel.Design.MenuCommand
```

Public Constructors

```
public DesignerVerb(
   System.String text,
   System.EventHandler handler
);
public DesignerVerb(
   System.String text,
   System.EventHandler handler,
   System.ComponentModel.Design.CommandID startCommandID
);
```

Public Properties

```
public System.String get_Text();
```

Public Methods

```
public System.String ToString();
```

DesignerVerbCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class DesignerVerbCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public DesignerVerbCollection();
public DesignerVerbCollection(
   System.ComponentModel.Design.DesignerVerb[] value
);
```

Public Properties

```
public System.ComponentModel.Design.DesignerVerb get_Item(
   int index
);
public void set_Item(
   int index,
   System.ComponentModel.Design.DesignerVerb value
);
```

Public Methods

```
public int Add(
   System.ComponentModel.Design.DesignerVerb value
);
public void AddRange(
   System.ComponentModel.Design.DesignerVerbCollection value
);
public void AddRange(
   System.ComponentModel.Design.DesignerVerb[] value
);
public boolean Contains(
   System.ComponentModel.Design.DesignerVerb value
);
public void CopyTo(
   System.ComponentModel.Design.DesignerVerb[] array,
   int index
);
public int IndexOf(
   System.ComponentModel.Design.DesignerVerb value
);
public void Insert(
   int index,
   System.ComponentModel.Design.DesignerVerb value
);
public void Remove(
   System.ComponentModel.Design.DesignerVerb value
);
```

Protected Methods

```
protected void OnClear();
protected void OnInsert(
   int index,
   System.Object value
);
protected void OnRemove(
   int index,
   System.Object value
);
protected void OnSet(
   int index,
   System.Object oldValue,
   System.Object newValue
);
protected void OnValidate(
   System.Object value
);
```

DesigntimeLicenseContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DesigntimeLicenseContext
   extends System.ComponentModel.LicenseContext
```

Public Constructors

```
public DesigntimeLicenseContext();
```

Public Properties

```
public System.ComponentModel.LicenseUsageMode get_UsageMode();
```

Public Methods

```
public System.String GetSavedLicenseKey(
   System.Type type,
   System.Reflection.Assembly resourceAssembly
);
public void SetSavedLicenseKey(
   System.Type type,
   System.String key
);
```

DesigntimeLicenseContextSerializer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DesigntimeLicenseContextSerializer
   extends System.Object
```

Public Methods

```
public static void Serialize(
   System.IO.Stream o,
   System.String cryptoKey,
   System.ComponentModel.Design.DesigntimeLicenseContext context
);
```

DisplayMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum DisplayMode
```

Members

Ansi

Auto

Hexdump

Unicode

HelpContextType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum HelpContextType
```

Members

Ambient

Selection

ToolWindowSelection

Window

HelpKeywordType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum HelpKeywordType
```

Members

F1Keyword

FilterKeyword

GeneralKeyword

IComponentChangeService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public interface IComponentChangeService
```

Public Methods

```
void OnComponentChanged(
   System.Object component,
   System.ComponentModel.MemberDescriptor member,
   System.Object oldValue,
   System.Object newValue
);
void OnComponentChanging(
   System.Object component,
   System.ComponentModel.MemberDescriptor member
);
```

Public Events

```
void add_ComponentAdded(
   System.ComponentModel.Design.ComponentEventHandler value
);
void remove_ComponentAdded(
   System.ComponentModel.Design.ComponentEventHandler value
);
void add_ComponentAdding(
   System.ComponentModel.Design.ComponentEventHandler value
);
void remove_ComponentAdding(
   System.ComponentModel.Design.ComponentEventHandler value
);
void add_ComponentChanged(
   System.ComponentModel.Design.ComponentChangedEventHandler value
);
void remove_ComponentChanged(
   System.ComponentModel.Design.ComponentChangedEventHandler value
);
void add_ComponentChanging(
   System.ComponentModel.Design.ComponentChangingEventHandler value
);
void remove_ComponentChanging(
   System.ComponentModel.Design.ComponentChangingEventHandler value
);
void add_ComponentRemoved(
   System.ComponentModel.Design.ComponentEventHandler value
);
void remove_ComponentRemoved(
   System.ComponentModel.Design.ComponentEventHandler value
);
void add_ComponentRemoving(
   System.ComponentModel.Design.ComponentEventHandler value
);
void remove_ComponentRemoving(
   System.ComponentModel.Design.ComponentEventHandler value
);
void add_ComponentRename(
```

```
    System.ComponentModel.Design.ComponentRenameEventHandler value
);
void remove_ComponentRename(
    System.ComponentModel.Design.ComponentRenameEventHandler value
);
```

IDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public interface IDesigner
   implements System.IDisposable
```

Public Properties

```
System.ComponentModel.IComponent get_Component();
System.ComponentModel.Design.DesignerVerbCollection get_Verbs();
```

Public Methods

```
void DoDefaultAction();
void Initialize(
   System.ComponentModel.IComponent component
);
```

IDesignerEventService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDesignerEventService
```

Public Properties

```
System.ComponentModel.Design.IDesignerHost get_ActiveDesigner();
System.ComponentModel.Design.DesignerCollection get_Designers();
```

Public Events

```
void add_ActiveDesignerChanged(
   System.ComponentModel.Design.ActiveDesignerEventHandler value
);
void remove_ActiveDesignerChanged(
   System.ComponentModel.Design.ActiveDesignerEventHandler value
);
void add_DesignerCreated(
   System.ComponentModel.Design.DesignerEventHandler value
);
void remove_DesignerCreated(
   System.ComponentModel.Design.DesignerEventHandler value
);
void add_DesignerDisposed(
   System.ComponentModel.Design.DesignerEventHandler value
);
void remove_DesignerDisposed(
   System.ComponentModel.Design.DesignerEventHandler value
);
void add_SelectionChanged(
   System.EventHandler value
);
void remove_SelectionChanged(
   System.EventHandler value
);
```

IDesignerFilter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDesignerFilter
```

Public Methods

```
void PostFilterAttributes(
   System.Collections.IDictionary attributes
);
void PostFilterEvents(
   System.Collections.IDictionary events
);
void PostFilterProperties(
   System.Collections.IDictionary properties
);
void PreFilterAttributes(
   System.Collections.IDictionary attributes
);
void PreFilterEvents(
   System.Collections.IDictionary events
);
void PreFilterProperties(
   System.Collections.IDictionary properties
);
```

IDesignerHost Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public interface IDesignerHost
   implements System.ComponentModel.Design.IServiceContainer, System.IServiceProvider
```

Public Properties

```
System.ComponentModel.IContainer get_Container();
boolean get_InTransaction();
boolean get_Loading();
System.ComponentModel.IComponent get_RootComponent();
System.String get_RootComponentClassName();
System.String get_TransactionDescription();
```

Public Methods

```
void Activate();
System.ComponentModel.IComponent CreateComponent(
   System.Type componentClass
);
System.ComponentModel.IComponent CreateComponent(
   System.Type componentClass,
   System.String name
);
System.ComponentModel.Design.DesignerTransaction CreateTransaction();
System.ComponentModel.Design.DesignerTransaction CreateTransaction(
   System.String description
);
void DestroyComponent(
   System.ComponentModel.IComponent component
);
System.ComponentModel.Design.IDesigner GetDesigner(
   System.ComponentModel.IComponent component
);
System.Type GetType(
   System.String typeName
);
```

Public Events

```
void add_Activated(
   System.EventHandler value
);
void remove_Activated(
   System.EventHandler value
);
void add_Deactivated(
   System.EventHandler value
);
void remove_Deactivated(
   System.EventHandler value
);
void add_LoadComplete(
   System.EventHandler value
```

```
);
void remove_LoadComplete(
   System.EventHandler value
);
void add_TransactionClosed(
   System.ComponentModel.Design.DesignerTransactionCloseEventHandler value
);
void remove_TransactionClosed(
   System.ComponentModel.Design.DesignerTransactionCloseEventHandler value
);
void add_TransactionClosing(
   System.ComponentModel.Design.DesignerTransactionCloseEventHandler value
);
void remove_TransactionClosing(
   System.ComponentModel.Design.DesignerTransactionCloseEventHandler value
);
void add_TransactionOpened(
   System.EventHandler value
);
void remove_TransactionOpened(
   System.EventHandler value
);
void add_TransactionOpening(
   System.EventHandler value
);
void remove_TransactionOpening(
   System.EventHandler value
);
```

IDesignerOptionService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDesignerOptionService
```

Public Methods

```
System.Object GetOptionValue(
   System.String pageName,
   System.String valueName
);
void SetOptionValue(
   System.String pageName,
   System.String valueName,
   System.Object value
);
```

IDictionaryService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDictionaryService
```

Public Methods

```
System.Object GetKey(
   System.Object value
);
System.Object GetValue(
   System.Object key
);
void SetValue(
   System.Object key,
   System.Object value
);
```

IEventBindingService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public interface IEventBindingService
```

Public Methods

```
System.String CreateUniqueMethodName(
   System.ComponentModel.IComponent component,
   System.ComponentModel.EventDescriptor e
);
System.Collections.ICollection GetCompatibleMethods(
   System.ComponentModel.EventDescriptor e
);
System.ComponentModel.EventDescriptor GetEvent(
   System.ComponentModel.PropertyDescriptor property
);
System.ComponentModel.PropertyDescriptorCollection GetEventProperties(
   System.ComponentModel.EventDescriptorCollection events
);
System.ComponentModel.PropertyDescriptor GetEventProperty(
   System.ComponentModel.EventDescriptor e
);
boolean ShowCode();
boolean ShowCode(
   int lineNumber
);
boolean ShowCode(
   System.ComponentModel.IComponent component,
   System.ComponentModel.EventDescriptor e
);
```

IExtenderListService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IExtenderListService
```

Public Methods

```
System.ComponentModel.IExtenderProvider[] GetExtenderProviders();
```

IExtenderProviderService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IExtenderProviderService
```

Public Methods

```
void AddExtenderProvider(
   System.ComponentModel.IExtenderProvider provider
);
void RemoveExtenderProvider(
   System.ComponentModel.IExtenderProvider provider
);
```

IHelpService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IHelpService
```

Public Methods

```
void AddContextAttribute(
   System.String name,
   System.String value,
   System.ComponentModel.Design.HelpKeywordType keywordType
);
void ClearContextAttributes();
System.ComponentModel.Design.IHelpService CreateLocalContext(
   System.ComponentModel.Design.HelpContextType contextType
);
void RemoveContextAttribute(
   System.String name,
   System.String value
);
void RemoveLocalContext(
   System.ComponentModel.Design.IHelpService localContext
);
void ShowHelpFromKeyword(
   System.String helpKeyword
);
void ShowHelpFromUrl(
   System.String helpUrl
);
```

IInheritanceService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IInheritanceService
```

Public Methods

```
void AddInheritedComponents(
   System.ComponentModel.IComponent component,
   System.ComponentModel.IContainer container
);
System.ComponentModel.InheritanceAttribute GetInheritanceAttribute(
   System.ComponentModel.IComponent component
);
```

IMenuCommandService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public interface IMenuCommandService
```

Public Properties

```
System.ComponentModel.Design.DesignerVerbCollection get_Verbs();
```

Public Methods

```
void AddCommand(
   System.ComponentModel.Design.MenuCommand command
);
void AddVerb(
   System.ComponentModel.Design.DesignerVerb verb
);
System.ComponentModel.Design.MenuCommand FindCommand(
   System.ComponentModel.Design.CommandID commandID
);
boolean GlobalInvoke(
   System.ComponentModel.Design.CommandID commandID
);
void RemoveCommand(
   System.ComponentModel.Design.MenuCommand command
);
void RemoveVerb(
   System.ComponentModel.Design.DesignerVerb verb
);
void ShowContextMenu(
   System.ComponentModel.Design.CommandID menuID,
   int x,
   int y
);
```

InheritanceService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InheritanceService
   extends System.Object
   implements System.ComponentModel.Design.IInheritanceService, System.IDisposable
```

Public Constructors

```
public InheritanceService();
```

Public Methods

```
public final void AddInheritedComponents(
   System.ComponentModel.IComponent component,
   System.ComponentModel.IContainer container
);
public final void Dispose();
public final System.ComponentModel.InheritanceAttribute GetInheritanceAttribute(
   System.ComponentModel.IComponent component
);
```

Protected Methods

```
protected void AddInheritedComponents(
   System.Type type,
   System.ComponentModel.IComponent component,
   System.ComponentModel.IContainer container
);
protected boolean IgnoreInheritedMember(
   System.Reflection.MemberInfo member,
   System.ComponentModel.IComponent component
);
```

IReferenceService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IReferenceService
```

Public Methods

```
System.ComponentModel.IComponent GetComponent(
   System.Object reference
);
System.String GetName(
   System.Object reference
);
System.Object GetReference(
   System.String name
);
System.Object[] GetReferences();
System.Object[] GetReferences(
   System.Type baseType
);
```

IResourceService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IResourceService
```

Public Methods

```
System.Resources.IResourceReader GetResourceReader(
   System.Globalization.CultureInfo info
);
System.Resources.IResourceWriter GetResourceWriter(
   System.Globalization.CultureInfo info
);
```

IRootDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public interface IRootDesigner
   implements System.ComponentModel.Design.IDesigner, System.IDisposable
```

Public Properties

```
System.ComponentModel.Design.ViewTechnology[] get_SupportedTechnologies();
```

Public Methods

```
System.Object GetView(
   System.ComponentModel.Design.ViewTechnology technology
);
```

ISelectionService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public interface ISelectionService
```

Public Properties

```
System.Object get_PrimarySelection();
int get_SelectionCount();
```

Public Methods

```
boolean GetComponentSelected(
   System.Object component
);
System.Collections.ICollection GetSelectedComponents();
void SetSelectedComponents(
   System.Collections.ICollection components
);
void SetSelectedComponents(
   System.Collections.ICollection components,
   System.ComponentModel.Design.SelectionTypes selectionType
);
```

Public Events

```
void add_SelectionChanged(
   System.EventHandler value
);
void remove_SelectionChanged(
   System.EventHandler value
);
void add_SelectionChanging(
   System.EventHandler value
);
void remove_SelectionChanging(
   System.EventHandler value
);
```

IServiceContainer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public interface IServiceContainer
   implements System.IServiceProvider
```

Public Methods

```
void AddService(
   System.Type serviceType,
   System.ComponentModel.Design.ServiceCreatorCallback callback
);
void AddService(
   System.Type serviceType,
   System.Object serviceInstance
);
void AddService(
   System.Type serviceType,
   System.ComponentModel.Design.ServiceCreatorCallback callback,
   boolean promote
);
void AddService(
   System.Type serviceType,
   System.Object serviceInstance,
   boolean promote
);
void RemoveService(
   System.Type serviceType
);
void RemoveService(
   System.Type serviceType,
   boolean promote
);
```

ITypeDescriptorFilterService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ITypeDescriptorFilterService
```

Public Methods

```
boolean FilterAttributes(
   System.ComponentModel.IComponent component,
   System.Collections.IDictionary attributes
);
boolean FilterEvents(
   System.ComponentModel.IComponent component,
   System.Collections.IDictionary events
);
boolean FilterProperties(
   System.ComponentModel.IComponent component,
   System.Collections.IDictionary properties
);
```

ITypeResolutionService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ITypeResolutionService
```

Public Methods

```
System.Reflection.Assembly GetAssembly(
   System.Reflection.AssemblyName name
);
System.Reflection.Assembly GetAssembly(
   System.Reflection.AssemblyName name,
   boolean throwOnError
);
System.String GetPathOfAssembly(
   System.Reflection.AssemblyName name
);
System.Type GetType(
   System.String name
);
System.Type GetType(
   System.String name,
   boolean throwOnError
);
System.Type GetType(
   System.String name,
   boolean throwOnError,
   boolean ignoreCase
);
void ReferenceAssembly(
   System.Reflection.AssemblyName name
);
```

LocalizationExtenderProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ProvidePropertyAttribute(PropertyName = "Localizable", R
eceiverTypeName = "System.Object, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyTok
en=b77a5c561934e089") */
/** @attribute System.ComponentModel.ProvidePropertyAttribute(PropertyName = "Language", Rece
iverTypeName = "System.Object, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=
b77a5c561934e089") */
/** @attribute System.ComponentModel.ProvidePropertyAttribute(PropertyName = "LoadLanguage",
ReceiverTypeName = "System.Object, mscorlib, Version=1.0.3300.0, Culture=neutral, PublicKeyTo
ken=b77a5c561934e089") */
public class LocalizationExtenderProvider
   extends System.Object
   implements System.ComponentModel.IExtenderProvider, System.IDisposable
```

Public Constructors

```
public LocalizationExtenderProvider(
   System.ComponentModel.ISite serviceProvider,
   System.ComponentModel.IComponent baseComponent
);
```

Public Methods

```
public final boolean CanExtend(
   System.Object o
);
public final void Dispose();
/** @attribute System.Design.SRDescriptionAttribute(Description = "Indicates the current loca
lizable language.") */
/** @attribute System.ComponentModel.LocalizableAttribute(IsLocalizable = true) */
/** @attribute System.ComponentModel.DesignOnlyAttribute(IsDesignOnly = true) */
public System.Globalization.CultureInfo GetLanguage(
   System.Object o
);
/** @attribute System.ComponentModel.DesignerSerializationVisibilityAttribute(Visibility = Hi
dden) */
/** @attribute System.ComponentModel.DesignOnlyAttribute(IsDesignOnly = true) */
/** @attribute System.ComponentModel.BrowsableAttribute(Browsable = false) */
public System.Globalization.CultureInfo GetLoadLanguage(
   System.Object o
);
/** @attribute System.ComponentModel.DesignOnlyAttribute(IsDesignOnly = true) */
/** @attribute System.Design.SRDescriptionAttribute(Description = "Determines if localizable
code will be generated for this object.") */
/** @attribute System.ComponentModel.DefaultValueAttribute(Value = False) */
/** @attribute System.ComponentModel.LocalizableAttribute(IsLocalizable = true) */
public boolean GetLocalizable(
   System.Object o
);
public void ResetLanguage(
   System.Object o
);
public void SetLanguage(
   System.Object o,
   System.Globalization.CultureInfo language
);
```

```
public void SetLocalizable(
   System.Object o,
   boolean localizable
);
public boolean ShouldSerializeLanguage(
   System.Object o
);
```

MenuCommand Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class MenuCommand
   extends System.Object
```

Public Constructors

```
public MenuCommand(
   System.EventHandler handler,
   System.ComponentModel.Design.CommandID command
);
```

Public Properties

```
public boolean get_Checked();
public void set_Checked(
   boolean value
);
public System.ComponentModel.Design.CommandID get_CommandID();
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public int get_OleStatus();
public boolean get_Supported();
public void set_Supported(
   boolean value
);
public boolean get_Visible();
public void set_Visible(
   boolean value
);
```

Public Methods

```
public void Invoke();
public System.String ToString();
```

Public Events

```
public void add_CommandChanged(
   System.EventHandler value
);
public void remove_CommandChanged(
   System.EventHandler value
);
```

Protected Methods

```
protected void OnCommandChanged(
```

```
    System.EventArgs e
);
```

ObjectSelectorEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class ObjectSelectorEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public ObjectSelectorEditor();
public ObjectSelectorEditor(
   boolean subObjectSelector
);
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public boolean EqualsToValue(
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
public void SetValue(
   System.Object value
);
```

Public Fields

```
public boolean SubObjectSelector;
```

Protected Methods

```
protected void FillTreeWithData(
   System.ComponentModel.Design.ObjectSelectorEditor+Selector selector,
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider
);
```

Protected Fields

```
protected System.Object currValue;
protected System.Object prevValue;
```

ObjectSelectorEditor.Selector Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Selector
   extends System.Windows.Forms.TreeView
```

Public Constructors

```
public Selector(
   System.ComponentModel.Design.ObjectSelectorEditor editor
);
```

Public Methods

```
public System.ComponentModel.Design.ObjectSelectorEditor+SelectorNode AddNode(
   System.String label,
   System.Object value,
   System.ComponentModel.Design.ObjectSelectorEditor+SelectorNode parent
);
public void Clear();
public boolean SetSelection(
   System.Object value,
   System.Windows.Forms.TreeNodeCollection nodes
);
public void Start(
   System.Windows.Forms.Design.IWindowsFormsEditorService edSvc,
   System.Object value
);
public void Stop();
```

Public Fields

```
public boolean clickSeen;
```

Protected Methods

```
protected void OnAfterSelect(
   System.Object sender,
   System.Windows.Forms.TreeViewEventArgs e
);
protected void OnKeyDown(
   System.Windows.Forms.KeyEventArgs e
);
protected void OnKeyPress(
   System.Windows.Forms.KeyPressEventArgs e
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

ObjectSelectorEditor.SelectorNode Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SelectorNode
   extends System.Windows.Forms.TreeNode
```

Public Constructors

```
public SelectorNode(
   System.String label,
   System.Object value
);
```

Public Fields

```
public System.Object value;
```

SelectionTypes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum SelectionTypes
```

Members

Click

MouseDown

MouseUp

Normal

Replace

Valid

ServiceContainer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ServiceContainer
   extends System.Object
   implements System.ComponentModel.Design.IServiceContainer, System.IServiceProvider
```

Public Constructors

```
public ServiceContainer();
public ServiceContainer(
   System.IServiceProvider parentProvider
);
```

Public Methods

```
public final void AddService(
   System.Type serviceType,
   System.ComponentModel.Design.ServiceCreatorCallback callback
);
public final void AddService(
   System.Type serviceType,
   System.Object serviceInstance
);
public final void AddService(
   System.Type serviceType,
   System.ComponentModel.Design.ServiceCreatorCallback callback,
   boolean promote
);
public final void AddService(
   System.Type serviceType,
   System.Object serviceInstance,
   boolean promote
);
public final System.Object GetService(
   System.Type serviceType
);
public final void RemoveService(
   System.Type serviceType
);
public final void RemoveService(
   System.Type serviceType,
   boolean promote
);
```

ServiceCreatorCallback Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
System.Object ServiceCreatorCallback(
   System.ComponentModel.Design.IServiceContainer container,
   System.Type serviceType
);
```

StandardCommands Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class StandardCommands
   extends System.Object
```

Public Constructors

```
public StandardCommands();
```

Public Fields

```
public static final System.ComponentModel.Design.CommandID AlignBottom;
public static final System.ComponentModel.Design.CommandID AlignHorizontalCenters;
public static final System.ComponentModel.Design.CommandID AlignLeft;
public static final System.ComponentModel.Design.CommandID AlignRight;
public static final System.ComponentModel.Design.CommandID AlignToGrid;
public static final System.ComponentModel.Design.CommandID AlignTop;
public static final System.ComponentModel.Design.CommandID AlignVerticalCenters;
public static final System.ComponentModel.Design.CommandID ArrangeBottom;
public static final System.ComponentModel.Design.CommandID ArrangeIcons;
public static final System.ComponentModel.Design.CommandID ArrangeRight;
public static final System.ComponentModel.Design.CommandID BringForward;
public static final System.ComponentModel.Design.CommandID BringToFront;
public static final System.ComponentModel.Design.CommandID CenterHorizontally;
public static final System.ComponentModel.Design.CommandID CenterVertically;
public static final System.ComponentModel.Design.CommandID Copy;
public static final System.ComponentModel.Design.CommandID Cut;
public static final System.ComponentModel.Design.CommandID Delete;
public static final System.ComponentModel.Design.CommandID F1Help;
public static final System.ComponentModel.Design.CommandID Group;
public static final System.ComponentModel.Design.CommandID HorizSpaceConcatenate;
public static final System.ComponentModel.Design.CommandID HorizSpaceDecrease;
public static final System.ComponentModel.Design.CommandID HorizSpaceIncrease;
public static final System.ComponentModel.Design.CommandID HorizSpaceMakeEqual;
public static final System.ComponentModel.Design.CommandID LineupIcons;
public static final System.ComponentModel.Design.CommandID LockControls;
public static final System.ComponentModel.Design.CommandID MultiLevelRedo;
public static final System.ComponentModel.Design.CommandID MultiLevelUndo;
public static final System.ComponentModel.Design.CommandID Paste;
public static final System.ComponentModel.Design.CommandID Properties;
public static final System.ComponentModel.Design.CommandID PropertiesWindow;
public static final System.ComponentModel.Design.CommandID Redo;
public static final System.ComponentModel.Design.CommandID Replace;
public static final System.ComponentModel.Design.CommandID SelectAll;
public static final System.ComponentModel.Design.CommandID SendBackward;
public static final System.ComponentModel.Design.CommandID SendToBack;
public static final System.ComponentModel.Design.CommandID ShowGrid;
public static final System.ComponentModel.Design.CommandID ShowLargeIcons;
public static final System.ComponentModel.Design.CommandID SizeToControl;
public static final System.ComponentModel.Design.CommandID SizeToControlHeight;
public static final System.ComponentModel.Design.CommandID SizeToControlWidth;
public static final System.ComponentModel.Design.CommandID SizeToFit;
public static final System.ComponentModel.Design.CommandID SizeToGrid;
public static final System.ComponentModel.Design.CommandID SnapToGrid;
public static final System.ComponentModel.Design.CommandID TabOrder;
public static final System.ComponentModel.Design.CommandID Undo;
public static final System.ComponentModel.Design.CommandID Ungroup;
```

```
public static final System.ComponentModel.Design.CommandID VerbFirst;
public static final System.ComponentModel.Design.CommandID VerbLast;
public static final System.ComponentModel.Design.CommandID VertSpaceConcatenate;
public static final System.ComponentModel.Design.CommandID VertSpaceDecrease;
public static final System.ComponentModel.Design.CommandID VertSpaceIncrease;
public static final System.ComponentModel.Design.CommandID VertSpaceMakeEqual;
public static final System.ComponentModel.Design.CommandID ViewGrid;
```

StandardToolWindows Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class StandardToolWindows
   extends System.Object
```

Public Constructors

```
public StandardToolWindows();
```

Public Fields

```
public static final System.Guid ObjectBrowser;
public static final System.Guid OutputWindow;
public static final System.Guid ProjectExplorer;
public static final System.Guid PropertyBrowser;
public static final System.Guid RelatedLinks;
public static final System.Guid ServerExplorer;
public static final System.Guid TaskList;
public static final System.Guid Toolbox;
```

ViewTechnology Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum ViewTechnology
```

Members

Passthrough

WindowsForms

System.ComponentModel.Design.Serialization Namespace (Visual J# Syntax)

Classes

CodeDomSerializer

CodeDomSerializerException

ContextStack

DesignerLoader

DesignerSerializerAttribute

InstanceDescriptor

ResolveNameEventArgs

RootDesignerSerializerAttribute

Interfaces

ICodeDomDesignerReload

IDesignerLoaderHost

IDesignerLoaderService

IDesignerSerializationManager

IDesignerSerializationProvider

IDesignerSerializationService

INameCreationService

Delegates

ResolveNameEventHandler

CodeDomSerializer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class CodeDomSerializer
   extends System.Object
```

Public Methods

```
public abstract System.Object Deserialize(
   System.ComponentModel.Design.Serialization.IDesignerSerializationManager manager,
   System.Object codeObject
);
public abstract System.Object Serialize(
   System.ComponentModel.Design.Serialization.IDesignerSerializationManager manager,
   System.Object value
);
```

Protected Constructors

```
protected CodeDomSerializer();
```

Protected Methods

```
protected System.Object DeserializeExpression(
   System.ComponentModel.Design.Serialization.IDesignerSerializationManager manager,
   System.String name,
   System.CodeDom.CodeExpression expression
);
protected void DeserializePropertiesFromResources(
   System.ComponentModel.Design.Serialization.IDesignerSerializationManager manager,
   System.Object value,
   System.Attribute[] filter
);
protected void DeserializeStatement(
   System.ComponentModel.Design.Serialization.IDesignerSerializationManager manager,
   System.CodeDom.CodeStatement statement
);
protected void SerializeEvents(
   System.ComponentModel.Design.Serialization.IDesignerSerializationManager manager,
   System.CodeDom.CodeStatementCollection statements,
   System.Object value,
   System.Attribute[] filter
);
protected void SerializeProperties(
   System.ComponentModel.Design.Serialization.IDesignerSerializationManager manager,
   System.CodeDom.CodeStatementCollection statements,
   System.Object value,
   System.Attribute[] filter
);
protected void SerializePropertiesToResources(
   System.ComponentModel.Design.Serialization.IDesignerSerializationManager manager,
   System.CodeDom.CodeStatementCollection statements,
   System.Object value,
   System.Attribute[] filter
);
protected void SerializeResource(
```

```
    System.ComponentModel.Design.Serialization.IDesignerSerializationManager manager,
    System.String resourceName,
    System.Object value
);
protected void SerializeResourceInvariant(
    System.ComponentModel.Design.Serialization.IDesignerSerializationManager manager,
    System.String resourceName,
    System.Object value
);
protected System.CodeDom.CodeExpression SerializeToExpression(
    System.ComponentModel.Design.Serialization.IDesignerSerializationManager manager,
    System.Object value
);
protected System.CodeDom.CodeExpression SerializeToReferenceExpression(
    System.ComponentModel.Design.Serialization.IDesignerSerializationManager manager,
    System.Object value
);
```

CodeDomSerializerException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CodeDomSerializerException
   extends System.SystemException
```

Public Constructors

```
public CodeDomSerializerException(
   System.Exception ex,
   System.CodeDom.CodeLinePragma linePragma
);
public CodeDomSerializerException(
   System.String message,
   System.CodeDom.CodeLinePragma linePragma
);
```

Public Properties

```
public System.CodeDom.CodeLinePragma get_LinePragma();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected CodeDomSerializerException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ContextStack Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ContextStack
   extends System.Object
```

Public Constructors

```
public ContextStack();
```

Public Properties

```
public System.Object get_Current();
public System.Object get_Item(
   int level
);
public System.Object get_Item(
   System.Type type
);
```

Public Methods

```
public System.Object Pop();
public void Push(
   System.Object context
);
```

DesignerLoader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class DesignerLoader
   extends System.Object
```

Public Properties

```
public boolean get_Loading();
```

Public Methods

```
public abstract void BeginLoad(
   System.ComponentModel.Design.Serialization.IDesignerLoaderHost host
);
public abstract void Dispose();
public void Flush();
```

Protected Constructors

```
protected DesignerLoader();
```

DesignerSerializerAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Interface, AllowMultiple = tru
e, Inherited = true) */
public class DesignerSerializerAttribute
   extends System.Attribute
```

Public Constructors

```
public DesignerSerializerAttribute(
   System.String serializerTypeName,
   System.String baseSerializerTypeName
);
public DesignerSerializerAttribute(
   System.String serializerTypeName,
   System.Type baseSerializerType
);
public DesignerSerializerAttribute(
   System.Type serializerType,
   System.Type baseSerializerType
);
```

Public Properties

```
public System.String get_SerializerBaseTypeName();
public System.String get_SerializerTypeName();
public System.Object get_TypeId();
```

ICodeDomDesignerReload Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICodeDomDesignerReload
```

Public Methods

```
boolean ShouldReloadDesigner(
   System.CodeDom.CodeCompileUnit newTree
);
```

IDesignerLoaderHost Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDesignerLoaderHost
   implements System.ComponentModel.Design.IDesignerHost, System.ComponentModel.Design.IServi
ceContainer, System.IServiceProvider
```

Public Methods

```
void EndLoad(
   System.String baseClassName,
   boolean successful,
   System.Collections.ICollection errorCollection
);
void Reload();
```

IDesignerLoaderService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDesignerLoaderService
```

Public Methods

```
void AddLoadDependency();
void DependentLoadComplete(
   boolean successful,
   System.Collections.ICollection errorCollection
);
boolean Reload();
```

IDesignerSerializationManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDesignerSerializationManager
   implements System.IServiceProvider
```

Public Properties

```
System.ComponentModel.Design.Serialization.ContextStack get_Context();
System.ComponentModel.PropertyDescriptorCollection get_Properties();
```

Public Methods

```
void AddSerializationProvider(
   System.ComponentModel.Design.Serialization.IDesignerSerializationProvider provider
);
System.Object CreateInstance(
   System.Type type,
   System.Collections.ICollection arguments,
   System.String name,
   boolean addToContainer
);
System.Object GetInstance(
   System.String name
);
System.String GetName(
   System.Object value
);
System.Object GetSerializer(
   System.Type objectType,
   System.Type serializerType
);
System.Type GetType(
   System.String typeName
);
void RemoveSerializationProvider(
   System.ComponentModel.Design.Serialization.IDesignerSerializationProvider provider
);
void ReportError(
   System.Object errorInformation
);
void SetName(
   System.Object instance,
   System.String name
);
```

Public Events

```
void add_ResolveName(
   System.ComponentModel.Design.Serialization.ResolveNameEventHandler value
);
void remove_ResolveName(
   System.ComponentModel.Design.Serialization.ResolveNameEventHandler value
);
void add_SerializationComplete(
   System.EventHandler value
```

```
);
void remove_SerializationComplete(
   System.EventHandler value
);
```

IDesignerSerializationProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDesignerSerializationProvider
```

Public Methods

```
System.Object GetSerializer(
   System.ComponentModel.Design.Serialization.IDesignerSerializationManager manager,
   System.Object currentSerializer,
   System.Type objectType,
   System.Type serializerType
);
```

IDesignerSerializationService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDesignerSerializationService
```

Public Methods

```
System.Collections.ICollection Deserialize(
   System.Object serializationData
);
System.Object Serialize(
   System.Collections.ICollection objects
);
```

INameCreationService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface INameCreationService
```

Public Methods

```
System.String CreateName(
   System.ComponentModel.IContainer container,
   System.Type dataType
);
boolean IsValidName(
   System.String name
);
void ValidateName(
   System.String name
);
```

InstanceDescriptor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InstanceDescriptor
   extends System.Object
```

Public Constructors

```
public InstanceDescriptor(
   System.Reflection.MemberInfo member,
   System.Collections.ICollection arguments
);
public InstanceDescriptor(
   System.Reflection.MemberInfo member,
   System.Collections.ICollection arguments,
   boolean isComplete
);
```

Public Properties

```
public System.Collections.ICollection get_Arguments();
public boolean get_IsComplete();
public System.Reflection.MemberInfo get_MemberInfo();
```

Public Methods

```
public System.Object Invoke();
```

ResolveNameEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ResolveNameEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ResolveNameEventArgs(
   System.String name
);
```

Public Properties

```
public System.String get_Name();
public System.Object get_Value();
public void set_Value(
   System.Object value
);
```

ResolveNameEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ResolveNameEventHandler(
   System.Object sender,
   System.ComponentModel.Design.Serialization.ResolveNameEventArgs e
);
```

RootDesignerSerializerAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Interface, AllowMultiple = tru
e, Inherited = true) */
public class RootDesignerSerializerAttribute
   extends System.Attribute
```

Public Constructors

```
public RootDesignerSerializerAttribute(
   System.String serializerTypeName,
   System.String baseSerializerTypeName,
   boolean reloadable
);
public RootDesignerSerializerAttribute(
   System.String serializerTypeName,
   System.Type baseSerializerType,
   boolean reloadable
);
public RootDesignerSerializerAttribute(
   System.Type serializerType,
   System.Type baseSerializerType,
   boolean reloadable
);
```

Public Properties

```
public boolean get_Reloadable();
public System.String get_SerializerBaseTypeName();
public System.String get_SerializerTypeName();
public System.Object get_TypeId();
```

System.Configuration Namespace (Visual J# Syntax)

Classes

AppSettingsReader

ConfigurationException

ConfigurationSettings

ConfigXmlDocument

DictionarySectionHandler

IgnoreSectionHandler

NameValueFileSectionHandler

NameValueSectionHandler

SingleTagSectionHandler

Interfaces

IConfigurationSectionHandler

IConfigurationSystem

AppSettingsReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AppSettingsReader
   extends System.Object
```

Public Constructors

```
public AppSettingsReader();
```

Public Methods

```
public System.Object GetValue(
   System.String key,
   System.Type type
);
```

ConfigurationException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ConfigurationException
   extends System.SystemException
```

Public Constructors

```
public ConfigurationException();
public ConfigurationException(
   System.String message
);
public ConfigurationException(
   System.String message,
   System.Exception inner
);
public ConfigurationException(
   System.String message,
   System.Xml.XmlNode node
);
public ConfigurationException(
   System.String message,
   System.Exception inner,
   System.Xml.XmlNode node
);
public ConfigurationException(
   System.String message,
   System.String filename,
   int line
);
public ConfigurationException(
   System.String message,
   System.Exception inner,
   System.String filename,
   int line
);
```

Public Properties

```
public System.String get_BareMessage();
public System.String get_Filename();
public int get_Line();
public System.String get_Message();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public static System.String GetXmlNodeFilename(
   System.Xml.XmlNode node
);
public static int GetXmlNodeLineNumber(
   System.Xml.XmlNode node
```

```
);
```

Protected Constructors

```
protected ConfigurationException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ConfigurationSettings Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ConfigurationSettings
   extends System.Object
```

Public Properties

```
public static System.Collections.Specialized.NameValueCollection get_AppSettings();
```

Public Methods

```
public static System.Object GetConfig(
   System.String sectionName
);
```

ConfigXmlDocument Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ConfigXmlDocument
   extends System.Xml.XmlDocument
```

Public Constructors

```
public ConfigXmlDocument();
```

Public Properties

```
public System.String get_Filename();
public int get_LineNumber();
```

Public Methods

```
public System.Xml.XmlAttribute CreateAttribute(
   System.String prefix,
   System.String localName,
   System.String namespaceUri
);
public System.Xml.XmlCDataSection CreateCDataSection(
   System.String data
);
public System.Xml.XmlComment CreateComment(
   System.String data
);
public System.Xml.XmlElement CreateElement(
   System.String prefix,
   System.String localName,
   System.String namespaceUri
);
public System.Xml.XmlSignificantWhitespace CreateSignificantWhitespace(
   System.String data
);
public System.Xml.XmlText CreateTextNode(
   System.String text
);
public System.Xml.XmlWhitespace CreateWhitespace(
   System.String data
);
public void Load(
   System.String filename
);
public void LoadSingleElement(
   System.String filename,
   System.Xml.XmlTextReader sourceReader
);
```

DictionarySectionHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DictionarySectionHandler
   extends System.Object
   implements System.Configuration.IConfigurationSectionHandler
```

Public Constructors

```
public DictionarySectionHandler();
```

Public Methods

```
public System.Object Create(
   System.Object parent,
   System.Object context,
   System.Xml.XmlNode section
);
```

Protected Properties

```
protected System.String get_KeyAttributeName();
protected System.String get_ValueAttributeName();
```

IConfigurationSectionHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IConfigurationSectionHandler
```

Public Methods

```
System.Object Create(
   System.Object parent,
   System.Object configContext,
   System.Xml.XmlNode section
);
```

IConfigurationSystem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IConfigurationSystem
```

Public Methods

```
System.Object GetConfig(
   System.String configKey
);
void Init();
```

IgnoreSectionHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class IgnoreSectionHandler
   extends System.Object
   implements System.Configuration.IConfigurationSectionHandler
```

Public Constructors

```
public IgnoreSectionHandler();
```

Public Methods

```
public System.Object Create(
   System.Object parent,
   System.Object configContext,
   System.Xml.XmlNode section
);
```

NameValueFileSectionHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class NameValueFileSectionHandler
   extends System.Object
   implements System.Configuration.IConfigurationSectionHandler
```

Public Constructors

```
public NameValueFileSectionHandler();
```

Public Methods

```
public final System.Object Create(
   System.Object parent,
   System.Object configContext,
   System.Xml.XmlNode section
);
```

NameValueSectionHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class NameValueSectionHandler
   extends System.Object
   implements System.Configuration.IConfigurationSectionHandler
```

Public Constructors

```
public NameValueSectionHandler();
```

Public Methods

```
public final System.Object Create(
   System.Object parent,
   System.Object context,
   System.Xml.XmlNode section
);
```

Protected Properties

```
protected System.String get_KeyAttributeName();
protected System.String get_ValueAttributeName();
```

SingleTagSectionHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SingleTagSectionHandler
   extends System.Object
   implements System.Configuration.IConfigurationSectionHandler
```

Public Constructors

```
public SingleTagSectionHandler();
```

Public Methods

```
public System.Object Create(
   System.Object parent,
   System.Object context,
   System.Xml.XmlNode section
);
```

System.Configuration.Assemblies Namespace (Visual J# Syntax)

Structures

AssemblyHash

Enumerations

AssemblyHashAlgorithm

AssemblyVersionCompatibility

AssemblyHash Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class AssemblyHash
   extends System.ValueType
   implements System.ICloneable
```

Public Constructors

```
public AssemblyHash(
   ubyte[] value
);
public AssemblyHash(
   System.Configuration.Assemblies.AssemblyHashAlgorithm algorithm,
   ubyte[] value
);
```

Public Properties

```
public System.Configuration.Assemblies.AssemblyHashAlgorithm get_Algorithm();
public void set_Algorithm(
   System.Configuration.Assemblies.AssemblyHashAlgorithm value
);
```

Public Methods

```
public final System.Object Clone();
public ubyte[] GetValue();
public void SetValue(
   ubyte[] value
);
```

Public Fields

```
public static final System.Configuration.Assemblies.AssemblyHash Empty;
```

AssemblyHashAlgorithm Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum AssemblyHashAlgorithm
```

Members

MD5

None

SHA1

AssemblyVersionCompatibility Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum AssemblyVersionCompatibility
```

Members

SameDomain

SameMachine

SameProcess

System.Configuration.Install Namespace (Visual J# Syntax)

Classes

AssemblyInstaller

ComponentInstaller

InstallContext

Installer

InstallerCollection

InstallEventArgs

InstallException

ManagedInstallerClass

TransactedInstaller

Interfaces

IManagedInstaller

Delegates

InstallEventHandler

Enumerations

UninstallAction

AssemblyInstaller Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AssemblyInstaller
   extends System.Configuration.Install.Installer
```

Public Constructors

```
public AssemblyInstaller();
public AssemblyInstaller(
   System.Reflection.Assembly assembly,
   System.String[] commandLine
);
public AssemblyInstaller(
   System.String filename,
   System.String[] commandLine
);
```

Public Properties

```
public System.Reflection.Assembly get_Assembly();
public void set_Assembly(
   System.Reflection.Assembly value
);
public System.String[] get_CommandLine();
public void set_CommandLine(
   System.String[] value
);
public System.String get_HelpText();
public System.String get_Path();
public void set_Path(
   System.String value
);
public boolean get_UseNewContext();
public void set_UseNewContext(
   boolean value
);
```

Public Methods

```
public static void CheckIfInstallable(
   System.String assemblyName
);
public void Commit(
   System.Collections.IDictionary savedState
);
public void Install(
   System.Collections.IDictionary savedState
);
public void Rollback(
   System.Collections.IDictionary savedState
);
public void Uninstall(
   System.Collections.IDictionary savedState
);
```

ComponentInstaller Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class ComponentInstaller
   extends System.Configuration.Install.Installer
```

Public Methods

```
public abstract void CopyFromComponent(
   System.ComponentModel.IComponent component
);
public boolean IsEquivalentInstaller(
   System.Configuration.Install.ComponentInstaller otherInstaller
);
```

Protected Constructors

```
protected ComponentInstaller();
```

IManagedInstaller Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "1E233FE7-C16D-4512-8C3B-
2E9988F08D38") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface IManagedInstaller
```

Public Methods

```
int ManagedInstall(
   System.String commandLine,
   int hInstall
);
```

InstallContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InstallContext
   extends System.Object
```

Public Constructors

```
public InstallContext();
public InstallContext(
   System.String logFilePath,
   System.String[] commandLine
);
```

Public Properties

```
public System.Collections.Specialized.StringDictionary get_Parameters();
```

Public Methods

```
public boolean IsParameterTrue(
   System.String paramName
);
public void LogMessage(
   System.String message
);
```

Protected Methods

```
protected static System.Collections.Specialized.StringDictionary ParseCommandLine(
   System.String[] args
);
```

Installer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "AfterInstall") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IRootDesigner, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77
a5c561934e089", DesignerTypeName = "Microsoft.VisualStudio.Configuration.InstallerDesigner, M
icrosoft.VisualStudio, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")
*/
public class Installer
   extends System.ComponentModel.Component
```

Public Constructors

```
public Installer();
```

Public Properties

```
public System.Configuration.Install.InstallContext get_Context();
public void set_Context(
   System.Configuration.Install.InstallContext value
);
public System.String get_HelpText();
public System.Configuration.Install.InstallerCollection get_Installers();
public System.Configuration.Install.Installer get_Parent();
public void set_Parent(
   System.Configuration.Install.Installer value
);
```

Public Methods

```
public void Commit(
   System.Collections.IDictionary savedState
);
public void Install(
   System.Collections.IDictionary stateSaver
);
public void Rollback(
   System.Collections.IDictionary savedState
);
public void Uninstall(
   System.Collections.IDictionary savedState
);
```

Public Events

```
public void add_AfterInstall(
   System.Configuration.Install.InstallEventHandler value
);
public void remove_AfterInstall(
   System.Configuration.Install.InstallEventHandler value
);
public void add_AfterRollback(
   System.Configuration.Install.InstallEventHandler value
```

```
);
public void remove_AfterRollback(
   System.Configuration.Install.InstallEventHandler value
);
public void add_AfterUninstall(
   System.Configuration.Install.InstallEventHandler value
);
public void remove_AfterUninstall(
   System.Configuration.Install.InstallEventHandler value
);
public void add_BeforeInstall(
   System.Configuration.Install.InstallEventHandler value
);
public void remove_BeforeInstall(
   System.Configuration.Install.InstallEventHandler value
);
public void add_BeforeRollback(
   System.Configuration.Install.InstallEventHandler value
);
public void remove_BeforeRollback(
   System.Configuration.Install.InstallEventHandler value
);
public void add_BeforeUninstall(
   System.Configuration.Install.InstallEventHandler value
);
public void remove_BeforeUninstall(
   System.Configuration.Install.InstallEventHandler value
);
public void add_Committed(
   System.Configuration.Install.InstallEventHandler value
);
public void remove_Committed(
   System.Configuration.Install.InstallEventHandler value
);
public void add_Committing(
   System.Configuration.Install.InstallEventHandler value
);
public void remove_Committing(
   System.Configuration.Install.InstallEventHandler value
);
```

Protected Methods

```
protected void OnAfterInstall(
   System.Collections.IDictionary savedState
);
protected void OnAfterRollback(
   System.Collections.IDictionary savedState
);
protected void OnAfterUninstall(
   System.Collections.IDictionary savedState
);
protected void OnBeforeInstall(
   System.Collections.IDictionary savedState
);
protected void OnBeforeRollback(
   System.Collections.IDictionary savedState
);
protected void OnBeforeUninstall(
   System.Collections.IDictionary savedState
);
protected void OnCommitted(
   System.Collections.IDictionary savedState
);
protected void OnCommitting(
   System.Collections.IDictionary savedState
```

```
);
```

InstallerCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class InstallerCollection
   extends System.Collections.CollectionBase
```

Public Properties

```
public System.Configuration.Install.Installer get_Item(
   int index
);
public void set_Item(
   int index,
   System.Configuration.Install.Installer value
);
```

Public Methods

```
public int Add(
   System.Configuration.Install.Installer value
);
public void AddRange(
   System.Configuration.Install.InstallerCollection value
);
public void AddRange(
   System.Configuration.Install.Installer[] value
);
public boolean Contains(
   System.Configuration.Install.Installer value
);
public void CopyTo(
   System.Configuration.Install.Installer[] array,
   int index
);
public int IndexOf(
   System.Configuration.Install.Installer value
);
public void Insert(
   int index,
   System.Configuration.Install.Installer value
);
public void Remove(
   System.Configuration.Install.Installer value
);
```

Protected Methods

```
protected void OnInsert(
   int index,
   System.Object value
);
protected void OnRemove(
   int index,
   System.Object value
);
```

```
protected void OnSet(
   int index,
   System.Object oldValue,
   System.Object newValue
);
```

InstallEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InstallEventArgs
   extends System.EventArgs
```

Public Constructors

```
public InstallEventArgs();
public InstallEventArgs(
   System.Collections.IDictionary savedState
);
```

Public Properties

```
public System.Collections.IDictionary get_SavedState();
```

InstallEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void InstallEventHandler(
   System.Object sender,
   System.Configuration.Install.InstallEventArgs e
);
```

InstallException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class InstallException
   extends System.SystemException
```

Public Constructors

```
public InstallException();
public InstallException(
   System.String message
);
public InstallException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected InstallException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ManagedInstallerClass Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "42EB0342-0393-448f-84AA-
D4BEB0283595") */
public class ManagedInstallerClass
   extends System.Object
   implements System.Configuration.Install.IManagedInstaller
```

Public Constructors

```
public ManagedInstallerClass();
```

Public Methods

```
public static void InstallHelper(
   System.String[] args
);
```

TransactedInstaller Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TransactedInstaller
   extends System.Configuration.Install.Installer
```

Public Constructors

```
public TransactedInstaller();
```

Public Methods

```
public void Install(
   System.Collections.IDictionary savedState
);
public void Uninstall(
   System.Collections.IDictionary savedState
);
```

UninstallAction Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum UninstallAction
```

Members

NoAction

Remove

System.Data Namespace (Visual J# Syntax)

Classes

Constraint

ConstraintCollection

ConstraintException

DataColumn

DataColumnChangeEventArgs

DataColumnCollection

DataException

DataRelation

DataRelationCollection

DataRow

DataRowBuilder

DataRowChangeEventArgs

DataRowCollection

DataRowView

DataSet

DataSysDescriptionAttribute

DataTable

DataTableCollection

DataView

DataViewManager

DataViewSetting

DataViewSettingCollection

DBConcurrencyException

DeletedRowInaccessibleException

DuplicateNameException

EvaluateException

FillErrorEventArgs

ForeignKeyConstraint

InRowChangingEventException

InternalDataCollectionBase

InvalidConstraintException

InvalidExpressionException

MergeFailedEventArgs

MissingPrimaryKeyException

NoNullAllowedException

PropertyCollection

ReadOnlyException

RowNotInTableException

StateChangeEventArgs

StrongTypingException

SyntaxErrorException

TypedDataSetGenerator

TypedDataSetGeneratorException

UniqueConstraint

VersionNotFoundException

Interfaces

IColumnMapping

IColumnMappingCollection

IDataAdapter

IDataParameter

IDataParameterCollection

IDataReader

IDataRecord

IDbCommand

IDbConnection

IDbDataAdapter

IDbDataParameter

IDbTransaction

ITableMapping

ITableMappingCollection

Delegates

DataColumnChangeEventHandler

DataRowChangeEventHandler

FillErrorEventHandler

MergeFailedEventHandler

StateChangeEventHandler

Enumerations

AcceptRejectRule

CommandBehavior

CommandType

ConnectionState

DataRowAction

DataRowState

DataRowVersion

DataViewRowState

DbType

IsolationLevel

MappingType

MissingMappingAction

MissingSchemaAction

ParameterDirection

PropertyAttributes

Rule

SchemaType

SqlDbType

StatementType

UpdateRowSource

UpdateStatus

XmlReadMode

XmlWriteMode

AcceptRejectRule Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum AcceptRejectRule
```

Members

Cascade

None

CommandBehavior Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum CommandBehavior
```

Members

CloseConnection

Default

KeyInfo

SchemaOnly

SequentialAccess

SingleResult

SingleRow

CommandType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CommandType
```

Members

StoredProcedure

TableDirect

Text

ConnectionState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum ConnectionState
```

Members

Broken

Closed

Connecting

Executing

Fetching

Open

Constraint Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "ConstraintName") */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Data.
ConstraintConverter, System.Data, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c5
61934e089") */
public abstract class Constraint
   extends System.Object
```

Public Properties

```
public System.String get_ConstraintName();
public void set_ConstraintName(
   System.String value
);
public System.Data.PropertyCollection get_ExtendedProperties();
public abstract System.Data.DataTable get_Table();
```

Public Methods

```
public System.String ToString();
```

Protected Constructors

```
protected Constraint();
```

Protected Properties

```
protected System.Data.DataSet get__DataSet();
```

Protected Methods

```
protected void CheckStateForProperty();
protected void SetDataSet(
   System.Data.DataSet dataSet
);
```

ConstraintCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.ConstraintsCollectionEditor, M
icrosoft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") *
/
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "CollectionChanged") */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ConstraintCollection
   extends System.Data.InternalDataCollectionBase
```

Public Properties

```
public System.Data.Constraint get_Item(
   int index
);
public System.Data.Constraint get_Item(
   System.String name
);
```

Public Methods

```
public void Add(
   System.Data.Constraint constraint
);
public System.Data.Constraint Add(
   System.String name,
   System.Data.DataColumn column,
   boolean primaryKey
);
public System.Data.Constraint Add(
   System.String name,
   System.Data.DataColumn primaryKeyColumn,
   System.Data.DataColumn foreignKeyColumn
);
public System.Data.Constraint Add(
   System.String name,
   System.Data.DataColumn[] columns,
   boolean primaryKey
);
public System.Data.Constraint Add(
   System.String name,
   System.Data.DataColumn[] primaryKeyColumns,
   System.Data.DataColumn[] foreignKeyColumns
);
public void AddRange(
   System.Data.Constraint[] constraints
);
public boolean CanRemove(
   System.Data.Constraint constraint
);
public void Clear();
public boolean Contains(
   System.String name
);
```

```
public int IndexOf(
   System.Data.Constraint constraint
);
public int IndexOf(
   System.String constraintName
);
public void Remove(
   System.Data.Constraint constraint
);
public void Remove(
   System.String name
);
public void RemoveAt(
   int index
);
```

Public Events

```
public void add_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
public void remove_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
```

Protected Properties

```
protected System.Collections.ArrayList get_List();
```

Protected Methods

```
protected void OnCollectionChanged(
   System.ComponentModel.CollectionChangeEventArgs ccevent
);
```

ConstraintException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ConstraintException
   extends System.Data.DataException
```

Public Constructors

```
public ConstraintException();
public ConstraintException(
   System.String s
);
```

Protected Constructors

```
protected ConstraintException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

DataColumn Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "ColumnName") */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.DataColumnEditor, Microsoft.VS
Designer, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class DataColumn
   extends System.ComponentModel.MarshalByValueComponent
```

Public Constructors

```
public DataColumn();
public DataColumn(
   System.String columnName
);
public DataColumn(
   System.String columnName,
   System.Type dataType
);
public DataColumn(
   System.String columnName,
   System.Type dataType,
   System.String expr
);
public DataColumn(
   System.String columnName,
   System.Type dataType,
   System.String expr,
   System.Data.MappingType type
);
```

Public Properties

```
public boolean get_AllowDBNull();
public void set_AllowDBNull(
   boolean value
);
public boolean get_AutoIncrement();
public void set_AutoIncrement(
   boolean value
);
public long get_AutoIncrementSeed();
public void set_AutoIncrementSeed(
   long value
);
public long get_AutoIncrementStep();
public void set_AutoIncrementStep(
   long value
);
public System.String get_Caption();
public void set_Caption(
   System.String value
);
```

```
public System.Data.MappingType get_ColumnMapping();
public void set_ColumnMapping(
   System.Data.MappingType value
);
public System.String get_ColumnName();
public void set_ColumnName(
   System.String value
);
public System.Type get_DataType();
public void set_DataType(
   System.Type value
);
public System.Object get_DefaultValue();
public void set_DefaultValue(
   System.Object value
);
public System.String get_Expression();
public void set_Expression(
   System.String value
);
public System.Data.PropertyCollection get_ExtendedProperties();
public int get_MaxLength();
public void set_MaxLength(
   int value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public int get_Ordinal();
public System.String get_Prefix();
public void set_Prefix(
   System.String value
);
public boolean get_ReadOnly();
public void set_ReadOnly(
   boolean value
);
public System.Data.DataTable get_Table();
public boolean get_Unique();
public void set_Unique(
   boolean value
);
```

Public Methods

```
public System.String ToString();
```

Protected Methods

```
protected void CheckNotAllowNull();
protected void CheckUnique();
protected void OnPropertyChanging(
   System.ComponentModel.PropertyChangedEventArgs pcevent
);
protected void RaisePropertyChanging(
   System.String name
);
```

DataColumnChangeEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataColumnChangeEventArgs
   extends System.EventArgs
```

Public Constructors

```
public DataColumnChangeEventArgs(
   System.Data.DataRow row,
   System.Data.DataColumn column,
   System.Object value
);
```

Public Properties

```
public System.Data.DataColumn get_Column();
public System.Object get_ProposedValue();
public void set_ProposedValue(
   System.Object value
);
public System.Data.DataRow get_Row();
```

DataColumnChangeEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DataColumnChangeEventHandler(
   System.Object sender,
   System.Data.DataColumnChangeEventArgs e
);
```

DataColumnCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "CollectionChanged") */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.ColumnsCollectionEditor, Micro
soft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class DataColumnCollection
   extends System.Data.InternalDataCollectionBase
```

Public Properties

```
public System.Data.DataColumn get_Item(
   int index
);
public System.Data.DataColumn get_Item(
   System.String name
);
```

Public Methods

```
public System.Data.DataColumn Add();
public System.Data.DataColumn Add(
   System.String columnName
);
public void Add(
   System.Data.DataColumn column
);
public System.Data.DataColumn Add(
   System.String columnName,
   System.Type type
);
public System.Data.DataColumn Add(
   System.String columnName,
   System.Type type,
   System.String expression
);
public void AddRange(
   System.Data.DataColumn[] columns
);
public boolean CanRemove(
   System.Data.DataColumn column
);
public void Clear();
public boolean Contains(
   System.String name
);
public int IndexOf(
   System.Data.DataColumn column
);
public int IndexOf(
   System.String columnName
);
public void Remove(
   System.Data.DataColumn column
```

```
);
public void Remove(
   System.String name
);
public void RemoveAt(
   int index
);
```

Public Events

```
public void add_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
public void remove_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
```

Protected Properties

```
protected System.Collections.ArrayList get_List();
```

Protected Methods

```
protected void OnCollectionChanged(
   System.ComponentModel.CollectionChangeEventArgs ccevent
);
protected void OnCollectionChanging(
   System.ComponentModel.CollectionChangeEventArgs ccevent
);
```

DataException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DataException
   extends System.SystemException
```

Public Constructors

```
public DataException();
public DataException(
   System.String s
);
public DataException(
   System.String s,
   System.Exception innerException
);
```

Protected Constructors

```
protected DataException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

DataRelation Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.DataRelationEditor, Microsoft.
VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Data.
RelationshipConverter, System.Data, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5
c561934e089") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "RelationName") */
public class DataRelation
   extends System.Object
```

Public Constructors

```
public DataRelation(
   System.String relationName,
   System.Data.DataColumn parentColumn,
   System.Data.DataColumn childColumn
);
public DataRelation(
   System.String relationName,
   System.Data.DataColumn[] parentColumns,
   System.Data.DataColumn[] childColumns
);
public DataRelation(
   System.String relationName,
   System.Data.DataColumn parentColumn,
   System.Data.DataColumn childColumn,
   boolean createConstraints
);
public DataRelation(
   System.String relationName,
   System.Data.DataColumn[] parentColumns,
   System.Data.DataColumn[] childColumns,
   boolean createConstraints
);
/** @attribute System.ComponentModel.BrowsableAttribute(Browsable = false) */
public DataRelation(
   System.String relationName,
   System.String parentTableName,
   System.String childTableName,
   System.String[] parentColumnNames,
   System.String[] childColumnNames,
   boolean nested
);
```

Public Properties

```
public System.Data.DataColumn[] get_ChildColumns();
public System.Data.ForeignKeyConstraint get_ChildKeyConstraint();
public System.Data.DataTable get_ChildTable();
public System.Data.DataSet get_DataSet();
public System.Data.PropertyCollection get_ExtendedProperties();
public boolean get_Nested();
public void set_Nested(
```

```
   boolean value
);
public System.Data.DataColumn[] get_ParentColumns();
public System.Data.UniqueConstraint get_ParentKeyConstraint();
public System.Data.DataTable get_ParentTable();
public System.String get_RelationName();
public void set_RelationName(
   System.String value
);
```

Public Methods

```
public System.String ToString();
```

Protected Methods

```
protected void CheckStateForProperty();
protected void OnPropertyChanging(
   System.ComponentModel.PropertyChangedEventArgs pcevent
);
protected void RaisePropertyChanging(
   System.String name
);
```

DataRelationCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "CollectionChanged") */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.DataRelationCollectionEditor,
Microsoft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")
*/
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Table") */
public abstract class DataRelationCollection
   extends System.Data.InternalDataCollectionBase
```

Public Properties

```
public abstract System.Data.DataRelation get_Item(
   int index
);
public abstract System.Data.DataRelation get_Item(
   System.String name
);
```

Public Methods

```
public void Add(
   System.Data.DataRelation relation
);
public System.Data.DataRelation Add(
   System.Data.DataColumn parentColumn,
   System.Data.DataColumn childColumn
);
public System.Data.DataRelation Add(
   System.Data.DataColumn[] parentColumns,
   System.Data.DataColumn[] childColumns
);
public System.Data.DataRelation Add(
   System.String name,
   System.Data.DataColumn parentColumn,
   System.Data.DataColumn childColumn
);
public System.Data.DataRelation Add(
   System.String name,
   System.Data.DataColumn[] parentColumns,
   System.Data.DataColumn[] childColumns
);
public System.Data.DataRelation Add(
   System.String name,
   System.Data.DataColumn parentColumn,
   System.Data.DataColumn childColumn,
   boolean createConstraints
);
public System.Data.DataRelation Add(
   System.String name,
   System.Data.DataColumn[] parentColumns,
   System.Data.DataColumn[] childColumns,
   boolean createConstraints
```

```
);
public void AddRange(
   System.Data.DataRelation[] relations
);
public boolean CanRemove(
   System.Data.DataRelation relation
);
public void Clear();
public boolean Contains(
   System.String name
);
public int IndexOf(
   System.Data.DataRelation relation
);
public int IndexOf(
   System.String relationName
);
public void Remove(
   System.Data.DataRelation relation
);
public void Remove(
   System.String name
);
public void RemoveAt(
   int index
);
```

Public Events

```
public void add_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
public void remove_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
```

Protected Constructors

```
protected DataRelationCollection();
```

Protected Methods

```
protected void AddCore(
   System.Data.DataRelation relation
);
protected abstract System.Data.DataSet GetDataSet();
protected void OnCollectionChanged(
   System.ComponentModel.CollectionChangeEventArgs ccevent
);
protected void OnCollectionChanging(
   System.ComponentModel.CollectionChangeEventArgs ccevent
);
protected void RemoveCore(
   System.Data.DataRelation relation
);
```

DataRow Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DataRow
   extends System.Object
```

Public Properties

```
public boolean get_HasErrors();
public System.Object get_Item(
   int columnIndex
);
public System.Object get_Item(
   System.Data.DataColumn column
);
public System.Object get_Item(
   System.String columnName
);
public System.Object get_Item(
   int columnIndex,
   System.Data.DataRowVersion version
);
public System.Object get_Item(
   System.Data.DataColumn column,
   System.Data.DataRowVersion version
);
public System.Object get_Item(
   System.String columnName,
   System.Data.DataRowVersion version
);
public void set_Item(
   int columnIndex,
   System.Object value
);
public void set_Item(
   System.Data.DataColumn column,
   System.Object value
);
public void set_Item(
   System.String columnName,
   System.Object value
);
public System.Object[] get_ItemArray();
public void set_ItemArray(
   System.Object[] value
);
public System.String get_RowError();
public void set_RowError(
   System.String value
);
public System.Data.DataRowState get_RowState();
public System.Data.DataTable get_Table();
```

Public Methods

```
public void AcceptChanges();
```

```
public void BeginEdit();
public void CancelEdit();
public void ClearErrors();
public void Delete();
public void EndEdit();
public System.Data.DataRow[] GetChildRows(
   System.Data.DataRelation relation
);
public System.Data.DataRow[] GetChildRows(
   System.String relationName
);
public System.Data.DataRow[] GetChildRows(
   System.Data.DataRelation relation,
   System.Data.DataRowVersion version
);
public System.Data.DataRow[] GetChildRows(
   System.String relationName,
   System.Data.DataRowVersion version
);
public System.String GetColumnError(
   int columnIndex
);
public System.String GetColumnError(
   System.Data.DataColumn column
);
public System.String GetColumnError(
   System.String columnName
);
public System.Data.DataColumn[] GetColumnsInError();
public System.Data.DataRow GetParentRow(
   System.Data.DataRelation relation
);
public System.Data.DataRow GetParentRow(
   System.String relationName
);
public System.Data.DataRow GetParentRow(
   System.Data.DataRelation relation,
   System.Data.DataRowVersion version
);
public System.Data.DataRow GetParentRow(
   System.String relationName,
   System.Data.DataRowVersion version
);
public System.Data.DataRow[] GetParentRows(
   System.Data.DataRelation relation
);
public System.Data.DataRow[] GetParentRows(
   System.String relationName
);
public System.Data.DataRow[] GetParentRows(
   System.Data.DataRelation relation,
   System.Data.DataRowVersion version
);
public System.Data.DataRow[] GetParentRows(
   System.String relationName,
   System.Data.DataRowVersion version
);
public boolean HasVersion(
   System.Data.DataRowVersion version
);
public boolean IsNull(
   int columnIndex
);
public boolean IsNull(
   System.Data.DataColumn column
);
public boolean IsNull(
   System.String columnName
```

```
);
public boolean IsNull(
   System.Data.DataColumn column,
   System.Data.DataRowVersion version
);
public void RejectChanges();
public void SetColumnError(
   int columnIndex,
   System.String error
);
public void SetColumnError(
   System.Data.DataColumn column,
   System.String error
);
public void SetColumnError(
   System.String columnName,
   System.String error
);
public void SetParentRow(
   System.Data.DataRow parentRow
);
public void SetParentRow(
   System.Data.DataRow parentRow,
   System.Data.DataRelation relation
);
```

Protected Constructors

```
protected DataRow(
   System.Data.DataRowBuilder builder
);
```

Protected Methods

```
protected void SetNull(
   System.Data.DataColumn column
);
```

DataRowAction Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum DataRowAction
```

Members

Add

Change

Commit

Delete

Nothing

Rollback

DataRowBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataRowBuilder
   extends System.Object
```

DataRowChangeEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataRowChangeEventArgs
   extends System.EventArgs
```

Public Constructors

```
public DataRowChangeEventArgs(
   System.Data.DataRow row,
   System.Data.DataRowAction action
);
```

Public Properties

```
public System.Data.DataRowAction get_Action();
public System.Data.DataRow get_Row();
```

DataRowChangeEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DataRowChangeEventHandler(
   System.Object sender,
   System.Data.DataRowChangeEventArgs e
);
```

DataRowCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DataRowCollection
   extends System.Data.InternalDataCollectionBase
```

Public Properties

```
public System.Data.DataRow get_Item(
   int index
);
```

Public Methods

```
public System.Data.DataRow Add(
   System.Object[] values
);
public void Add(
   System.Data.DataRow row
);
public void Clear();
public boolean Contains(
   System.Object key
);
public boolean Contains(
   System.Object[] keys
);
public System.Data.DataRow Find(
   System.Object key
);
public System.Data.DataRow Find(
   System.Object[] keys
);
public void InsertAt(
   System.Data.DataRow row,
   int pos
);
public void Remove(
   System.Data.DataRow row
);
public void RemoveAt(
   int index
);
```

Protected Properties

```
protected System.Collections.ArrayList get_List();
```

DataRowState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum DataRowState
```

Members

Added

Deleted

Detached

Modified

Unchanged

DataRowVersion Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum DataRowVersion
```

Members

Current

Default

Original

Proposed

DataRowView Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DataRowView
   extends System.Object
   implements System.ComponentModel.ICustomTypeDescriptor, System.ComponentModel.IEditableObj
ect, System.ComponentModel.IDataErrorInfo
```

Public Properties

```
public System.Data.DataView get_DataView();
public boolean get_IsEdit();
public boolean get_IsNew();
public System.Object get_Item(
   int ndx
);
public System.Object get_Item(
   System.String property
);
public void set_Item(
   int ndx,
   System.Object value
);
public void set_Item(
   System.String property,
   System.Object value
);
public System.Data.DataRow get_Row();
public System.Data.DataRowVersion get_RowVersion();
```

Public Methods

```
public final void BeginEdit();
public final void CancelEdit();
public System.Data.DataView CreateChildView(
   System.Data.DataRelation relation
);
public System.Data.DataView CreateChildView(
   System.String relationName
);
public void Delete();
public final void EndEdit();
public boolean Equals(
   System.Object other
);
public int GetHashCode();
```

DataSet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "Microsoft.VS
Designer.Data.VS.DataSetToolboxItem, Microsoft.VSDesigner, Version=7.0.3300.0, Culture=neutra
l, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "Microsoft.VSDesigner.Data.VS.DataSetDesigner,
Microsoft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")
*/
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "DataSetName") */
public class DataSet
   extends System.ComponentModel.MarshalByValueComponent
   implements System.ComponentModel.IListSource, System.Xml.Serialization.IXmlSerializable, S
ystem.ComponentModel.ISupportInitialize, System.Runtime.Serialization.ISerializable
```

Public Constructors

```
public DataSet();
public DataSet(
   System.String dataSetName
);
```

Public Properties

```
public boolean get_CaseSensitive();
public void set_CaseSensitive(
   boolean value
);
public System.String get_DataSetName();
public void set_DataSetName(
   System.String value
);
public System.Data.DataViewManager get_DefaultViewManager();
public boolean get_EnforceConstraints();
public void set_EnforceConstraints(
   boolean value
);
public System.Data.PropertyCollection get_ExtendedProperties();
public boolean get_HasErrors();
public System.Globalization.CultureInfo get_Locale();
public void set_Locale(
   System.Globalization.CultureInfo value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.String get_Prefix();
public void set_Prefix(
   System.String value
);
public System.Data.DataRelationCollection get_Relations();
public System.ComponentModel.ISite get_Site();
public void set_Site(
   System.ComponentModel.ISite value
);
```

```
public System.Data.DataTableCollection get_Tables();
```

Public Methods

```
public void AcceptChanges();
public final void BeginInit();
public void Clear();
public System.Data.DataSet Clone();
public System.Data.DataSet Copy();
public final void EndInit();
public System.Data.DataSet GetChanges();
public System.Data.DataSet GetChanges(
   System.Data.DataRowState rowStates
);
public System.String GetXml();
public System.String GetXmlSchema();
public boolean HasChanges();
public boolean HasChanges(
   System.Data.DataRowState rowStates
);
public void InferXmlSchema(
   System.IO.Stream stream,
   System.String[] nsArray
);
public void InferXmlSchema(
   System.IO.TextReader reader,
   System.String[] nsArray
);
public void InferXmlSchema(
   System.String fileName,
   System.String[] nsArray
);
public void InferXmlSchema(
   System.Xml.XmlReader reader,
   System.String[] nsArray
);
public void Merge(
   System.Data.DataRow[] rows
);
public void Merge(
   System.Data.DataSet dataSet
);
public void Merge(
   System.Data.DataTable table
);
public void Merge(
   System.Data.DataSet dataSet,
   boolean preserveChanges
);
public void Merge(
   System.Data.DataRow[] rows,
   boolean preserveChanges,
   System.Data.MissingSchemaAction missingSchemaAction
);
public void Merge(
   System.Data.DataSet dataSet,
   boolean preserveChanges,
   System.Data.MissingSchemaAction missingSchemaAction
);
public void Merge(
   System.Data.DataTable table,
   boolean preserveChanges,
   System.Data.MissingSchemaAction missingSchemaAction
);
public System.Data.XmlReadMode ReadXml(
   System.IO.Stream stream
```

```
);
public System.Data.XmlReadMode ReadXml(
   System.IO.TextReader reader
);
public System.Data.XmlReadMode ReadXml(
   System.String fileName
);
public System.Data.XmlReadMode ReadXml(
   System.Xml.XmlReader reader
);
public System.Data.XmlReadMode ReadXml(
   System.IO.Stream stream,
   System.Data.XmlReadMode mode
);
public System.Data.XmlReadMode ReadXml(
   System.IO.TextReader reader,
   System.Data.XmlReadMode mode
);
public System.Data.XmlReadMode ReadXml(
   System.String fileName,
   System.Data.XmlReadMode mode
);
public System.Data.XmlReadMode ReadXml(
   System.Xml.XmlReader reader,
   System.Data.XmlReadMode mode
);
public void ReadXmlSchema(
   System.IO.Stream stream
);
public void ReadXmlSchema(
   System.IO.TextReader reader
);
public void ReadXmlSchema(
   System.String fileName
);
public void ReadXmlSchema(
   System.Xml.XmlReader reader
);
public void RejectChanges();
public void Reset();
public void WriteXml(
   System.IO.Stream stream
);
public void WriteXml(
   System.IO.TextWriter writer
);
public void WriteXml(
   System.String fileName
);
public void WriteXml(
   System.Xml.XmlWriter writer
);
public void WriteXml(
   System.IO.Stream stream,
   System.Data.XmlWriteMode mode
);
public void WriteXml(
   System.IO.TextWriter writer,
   System.Data.XmlWriteMode mode
);
public void WriteXml(
   System.String fileName,
   System.Data.XmlWriteMode mode
);
public void WriteXml(
   System.Xml.XmlWriter writer,
   System.Data.XmlWriteMode mode
);
```

```
public void WriteXmlSchema(
   System.IO.Stream stream
);
public void WriteXmlSchema(
   System.IO.TextWriter writer
);
public void WriteXmlSchema(
   System.String fileName
);
public void WriteXmlSchema(
   System.Xml.XmlWriter writer
);
```

Public Events

```
public void add_MergeFailed(
   System.Data.MergeFailedEventHandler value
);
public void remove_MergeFailed(
   System.Data.MergeFailedEventHandler value
);
```

Protected Constructors

```
protected DataSet(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Methods

```
protected System.Xml.Schema.XmlSchema GetSchemaSerializable();
protected void GetSerializationData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
protected void OnPropertyChanging(
   System.ComponentModel.PropertyChangedEventArgs pcevent
);
protected void OnRemoveRelation(
   System.Data.DataRelation relation
);
protected void OnRemoveTable(
   System.Data.DataTable table
);
protected void RaisePropertyChanging(
   System.String name
);
protected void ReadXmlSerializable(
   System.Xml.XmlReader reader
);
protected boolean ShouldSerializeRelations();
protected boolean ShouldSerializeTables();
```

DataSysDescriptionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class DataSysDescriptionAttribute
   extends System.ComponentModel.DescriptionAttribute
```

Public Constructors

```
public DataSysDescriptionAttribute(
   System.String description
);
```

Public Properties

```
public System.String get_Description();
```

DataTable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "TableName") */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.DataTableEditor, Microsoft.VSD
esigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "RowChanging") */
public class DataTable
   extends System.ComponentModel.MarshalByValueComponent
   implements System.ComponentModel.IListSource, System.ComponentModel.ISupportInitialize, Sy
stem.Runtime.Serialization.ISerializable
```

Public Constructors

```
public DataTable();
public DataTable(
   System.String tableName
);
```

Public Properties

```
public boolean get_CaseSensitive();
public void set_CaseSensitive(
   boolean value
);
public System.Data.DataRelationCollection get_ChildRelations();
public System.Data.DataColumnCollection get_Columns();
public System.Data.ConstraintCollection get_Constraints();
public System.Data.DataSet get_DataSet();
public System.Data.DataView get_DefaultView();
public System.String get_DisplayExpression();
public void set_DisplayExpression(
   System.String value
);
public System.Data.PropertyCollection get_ExtendedProperties();
public boolean get_HasErrors();
public System.Globalization.CultureInfo get_Locale();
public void set_Locale(
   System.Globalization.CultureInfo value
);
public int get_MinimumCapacity();
public void set_MinimumCapacity(
   int value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.Data.DataRelationCollection get_ParentRelations();
public System.String get_Prefix();
public void set_Prefix(
   System.String value
);
```

```
public System.Data.DataColumn[] get_PrimaryKey();
public void set_PrimaryKey(
   System.Data.DataColumn[] value
);
public System.Data.DataRowCollection get_Rows();
public System.ComponentModel.ISite get_Site();
public void set_Site(
   System.ComponentModel.ISite value
);
public System.String get_TableName();
public void set_TableName(
   System.String value
);
```

Public Methods

```
public void AcceptChanges();
public final void BeginInit();
public void BeginLoadData();
public void Clear();
public System.Data.DataTable Clone();
public System.Object Compute(
   System.String expression,
   System.String filter
);
public System.Data.DataTable Copy();
public final void EndInit();
public void EndLoadData();
public System.Data.DataTable GetChanges();
public System.Data.DataTable GetChanges(
   System.Data.DataRowState rowStates
);
public System.Data.DataRow[] GetErrors();
public void ImportRow(
   System.Data.DataRow row
);
public System.Data.DataRow LoadDataRow(
   System.Object[] values,
   boolean fAcceptChanges
);
public System.Data.DataRow NewRow();
public void RejectChanges();
public void Reset();
public System.Data.DataRow[] Select();
public System.Data.DataRow[] Select(
   System.String filterExpression
);
public System.Data.DataRow[] Select(
   System.String filterExpression,
   System.String sort
);
public System.Data.DataRow[] Select(
   System.String filterExpression,
   System.String sort,
   System.Data.DataViewRowState recordStates
);
public System.String ToString();
```

Public Events

```
public void add_ColumnChanged(
   System.Data.DataColumnChangeEventHandler value
);
public void remove_ColumnChanged(
```

```
   System.Data.DataColumnChangeEventHandler value
);
public void add_ColumnChanging(
   System.Data.DataColumnChangeEventHandler value
);
public void remove_ColumnChanging(
   System.Data.DataColumnChangeEventHandler value
);
public void add_RowChanged(
   System.Data.DataRowChangeEventHandler value
);
public void remove_RowChanged(
   System.Data.DataRowChangeEventHandler value
);
public void add_RowChanging(
   System.Data.DataRowChangeEventHandler value
);
public void remove_RowChanging(
   System.Data.DataRowChangeEventHandler value
);
public void add_RowDeleted(
   System.Data.DataRowChangeEventHandler value
);
public void remove_RowDeleted(
   System.Data.DataRowChangeEventHandler value
);
public void add_RowDeleting(
   System.Data.DataRowChangeEventHandler value
);
public void remove_RowDeleting(
   System.Data.DataRowChangeEventHandler value
);
```

Protected Constructors

```
protected DataTable(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Methods

```
protected System.Data.DataTable CreateInstance();
protected System.Type GetRowType();
protected System.Data.DataRow[] NewRowArray(
   int size
);
protected System.Data.DataRow NewRowFromBuilder(
   System.Data.DataRowBuilder builder
);
protected void OnColumnChanged(
   System.Data.DataColumnChangeEventArgs e
);
protected void OnColumnChanging(
   System.Data.DataColumnChangeEventArgs e
);
protected void OnPropertyChanging(
   System.ComponentModel.PropertyChangedEventArgs pcevent
);
protected void OnRemoveColumn(
   System.Data.DataColumn column
);
protected void OnRowChanged(
   System.Data.DataRowChangeEventArgs e
```

```
);
protected void OnRowChanging(
   System.Data.DataRowChangeEventArgs e
);
protected void OnRowDeleted(
   System.Data.DataRowChangeEventArgs e
);
protected void OnRowDeleting(
   System.Data.DataRowChangeEventArgs e
);
```

Protected Fields

```
protected boolean fInitInProgress;
```

DataTableCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.ListBindableAttribute(ListBindable = false) */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.TablesCollectionEditor, Micros
oft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "CollectionChanged") */
public class DataTableCollection
   extends System.Data.InternalDataCollectionBase
```

Public Properties

```
public System.Data.DataTable get_Item(
   int index
);
public System.Data.DataTable get_Item(
   System.String name
);
```

Public Methods

```
public System.Data.DataTable Add();
public System.Data.DataTable Add(
   System.String name
);
public void Add(
   System.Data.DataTable table
);
public void AddRange(
   System.Data.DataTable[] tables
);
public boolean CanRemove(
   System.Data.DataTable table
);
public void Clear();
public boolean Contains(
   System.String name
);
public int IndexOf(
   System.Data.DataTable table
);
public int IndexOf(
   System.String tableName
);
public void Remove(
   System.Data.DataTable table
);
public void Remove(
   System.String name
);
public void RemoveAt(
   int index
);
```

Public Events

```
public void add_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
public void remove_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
public void add_CollectionChanging(
   System.ComponentModel.CollectionChangeEventHandler value
);
public void remove_CollectionChanging(
   System.ComponentModel.CollectionChangeEventHandler value
);
```

Protected Properties

```
protected System.Collections.ArrayList get_List();
```

Protected Methods

```
protected void OnCollectionChanged(
   System.ComponentModel.CollectionChangeEventArgs ccevent
);
protected void OnCollectionChanging(
   System.ComponentModel.CollectionChangeEventArgs ccevent
);
```

DataView Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "Microsoft.VSDesigner.Data.VS.DataViewDesigner,
Microsoft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")
*/
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Table") */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.DataSourceEditor, Microsoft.VS
Designer, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "PositionChanged") */
public class DataView
   extends System.ComponentModel.MarshalByValueComponent
   implements System.ComponentModel.IBindingList, System.Collections.IList, System.Collection
s.ICollection, System.Collections.IEnumerable, System.ComponentModel.ITypedList, System.Compo
nentModel.ISupportInitialize
```

Public Constructors

```
public DataView();
public DataView(
   System.Data.DataTable table
);
public DataView(
   System.Data.DataTable table,
   System.String RowFilter,
   System.String Sort,
   System.Data.DataViewRowState RowState
);
```

Public Properties

```
public boolean get_AllowDelete();
public void set_AllowDelete(
   boolean value
);
public boolean get_AllowEdit();
public void set_AllowEdit(
   boolean value
);
public boolean get_AllowNew();
public void set_AllowNew(
   boolean value
);
public boolean get_ApplyDefaultSort();
public void set_ApplyDefaultSort(
   boolean value
);
public final int get_Count();
public System.Data.DataViewManager get_DataViewManager();
public System.Data.DataRowView get_Item(
   int recordIndex
);
public System.String get_RowFilter();
public void set_RowFilter(
```

```
   System.String value
);
public System.Data.DataViewRowState get_RowStateFilter();
public void set_RowStateFilter(
   System.Data.DataViewRowState value
);
public System.String get_Sort();
public void set_Sort(
   System.String value
);
public System.Data.DataTable get_Table();
public void set_Table(
   System.Data.DataTable value
);
```

Public Methods

```
public System.Data.DataRowView AddNew();
public final void BeginInit();
public final void CopyTo(
   System.Array array,
   int index
);
public void Delete(
   int index
);
public final void EndInit();
public int Find(
   System.Object key
);
public int Find(
   System.Object[] key
);
public System.Data.DataRowView[] FindRows(
   System.Object key
);
public System.Data.DataRowView[] FindRows(
   System.Object[] key
);
public final System.Collections.IEnumerator GetEnumerator();
```

Public Events

```
public final void add_ListChanged(
   System.ComponentModel.ListChangedEventHandler value
);
public final void remove_ListChanged(
   System.ComponentModel.ListChangedEventHandler value
);
```

Protected Properties

```
protected boolean get_IsOpen();
```

Protected Methods

```
protected void Close();
protected void ColumnCollectionChanged(
   System.Object sender,
   System.ComponentModel.CollectionChangeEventArgs e
```

```
);
protected void Dispose(
   boolean disposing
);
protected void IndexListChanged(
   System.Object sender,
   System.ComponentModel.ListChangedEventArgs e
);
protected void OnListChanged(
   System.ComponentModel.ListChangedEventArgs e
);
protected void Open();
protected void Reset();
protected void UpdateIndex();
protected void UpdateIndex(
   boolean force
);
```

DataViewManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "Microsoft.VSDesigner.Data.VS.DataViewManagerDe
signer, Microsoft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d
50a3a") */
public class DataViewManager
   extends System.ComponentModel.MarshalByValueComponent
   implements System.ComponentModel.IBindingList, System.Collections.IList, System.Collection
s.ICollection, System.Collections.IEnumerable, System.ComponentModel.ITypedList
```

Public Constructors

```
public DataViewManager();
public DataViewManager(
   System.Data.DataSet dataSet
);
```

Public Properties

```
public System.Data.DataSet get_DataSet();
public void set_DataSet(
   System.Data.DataSet value
);
public System.String get_DataViewSettingCollectionString();
public void set_DataViewSettingCollectionString(
   System.String value
);
public System.Data.DataViewSettingCollection get_DataViewSettings();
```

Public Methods

```
public System.Data.DataView CreateDataView(
   System.Data.DataTable table
);
```

Public Events

```
public final void add_ListChanged(
   System.ComponentModel.ListChangedEventHandler value
);
public final void remove_ListChanged(
   System.ComponentModel.ListChangedEventHandler value
);
```

Protected Methods

```
protected void OnListChanged(
   System.ComponentModel.ListChangedEventArgs e
);
protected void RelationCollectionChanged(
```

```
   System.Object sender,
   System.ComponentModel.CollectionChangeEventArgs e
);
protected void TableCollectionChanged(
   System.Object sender,
   System.ComponentModel.CollectionChangeEventArgs e
);
```

DataViewRowState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.DataViewRowStateEditor, Micros
oft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public enum DataViewRowState
```

Members

Added

CurrentRows

Deleted

ModifiedCurrent

ModifiedOriginal

None

OriginalRows

Unchanged

DataViewSetting Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Compo
nentModel.ExpandableObjectConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyTo
ken=b77a5c561934e089") */
public class DataViewSetting
   extends System.Object
```

Public Properties

```
public boolean get_ApplyDefaultSort();
public void set_ApplyDefaultSort(
   boolean value
);
public System.Data.DataViewManager get_DataViewManager();
public System.String get_RowFilter();
public void set_RowFilter(
   System.String value
);
public System.Data.DataViewRowState get_RowStateFilter();
public void set_RowStateFilter(
   System.Data.DataViewRowState value
);
public System.String get_Sort();
public void set_Sort(
   System.String value
);
public System.Data.DataTable get_Table();
```

DataViewSettingCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.DataViewSettingsCollectionEdit
or, Microsoft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3
a") */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DataViewSettingCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Data.DataViewSetting get_Item(
   int index
);
public System.Data.DataViewSetting get_Item(
   System.Data.DataTable table
);
public System.Data.DataViewSetting get_Item(
   System.String tableName
);
public void set_Item(
   int index,
   System.Data.DataViewSetting value
);
public void set_Item(
   System.Data.DataTable table,
   System.Data.DataViewSetting value
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public final void CopyTo(
   System.Array ar,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
```

DBConcurrencyException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DBConcurrencyException
   extends System.SystemException
```

Public Constructors

```
public DBConcurrencyException(
   System.String message
);
public DBConcurrencyException(
   System.String message,
   System.Exception inner
);
```

Public Properties

```
public System.Data.DataRow get_Row();
public void set_Row(
   System.Data.DataRow value
);
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo si,
   System.Runtime.Serialization.StreamingContext context
);
```

DbType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum DbType
```

Members

AnsiString

AnsiStringFixedLength

Binary

Boolean

Byte

Currency

Date

DateTime

Decimal

Double

Guid

Int16

Int32

Int64

Object

SByte

Single

String

StringFixedLength

Time

UInt16

UInt32

UInt64

VarNumeric

DeletedRowInaccessibleException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DeletedRowInaccessibleException
   extends System.Data.DataException
```

Public Constructors

```
public DeletedRowInaccessibleException();
public DeletedRowInaccessibleException(
   System.String s
);
```

Protected Constructors

```
protected DeletedRowInaccessibleException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

DuplicateNameException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DuplicateNameException
   extends System.Data.DataException
```

Public Constructors

```
public DuplicateNameException();
public DuplicateNameException(
   System.String s
);
```

Protected Constructors

```
protected DuplicateNameException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

EvaluateException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class EvaluateException
   extends System.Data.InvalidExpressionException
```

Public Constructors

```
public EvaluateException();
public EvaluateException(
   System.String s
);
```

Protected Constructors

```
protected EvaluateException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

FillErrorEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FillErrorEventArgs
   extends System.EventArgs
```

Public Constructors

```
public FillErrorEventArgs(
   System.Data.DataTable dataTable,
   System.Object[] values
);
```

Public Properties

```
public boolean get_Continue();
public void set_Continue(
   boolean value
);
public System.Data.DataTable get_DataTable();
public System.Exception get_Errors();
public void set_Errors(
   System.Exception value
);
public System.Object[] get_Values();
```

FillErrorEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void FillErrorEventHandler(
   System.Object sender,
   System.Data.FillErrorEventArgs e
);
```

ForeignKeyConstraint Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "ConstraintName") */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.ForeignKeyConstraintEditor, Mi
crosoft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class ForeignKeyConstraint
   extends System.Data.Constraint
```

Public Constructors

```
public ForeignKeyConstraint(
   System.Data.DataColumn parentColumn,
   System.Data.DataColumn childColumn
);
public ForeignKeyConstraint(
   System.Data.DataColumn[] parentColumns,
   System.Data.DataColumn[] childColumns
);
public ForeignKeyConstraint(
   System.String constraintName,
   System.Data.DataColumn parentColumn,
   System.Data.DataColumn childColumn
);
public ForeignKeyConstraint(
   System.String constraintName,
   System.Data.DataColumn[] parentColumns,
   System.Data.DataColumn[] childColumns
);
/** @attribute System.ComponentModel.BrowsableAttribute(Browsable = false) */
public ForeignKeyConstraint(
   System.String constraintName,
   System.String parentTableName,
   System.String[] parentColumnNames,
   System.String[] childColumnNames,
   System.Data.AcceptRejectRule acceptRejectRule,
   System.Data.Rule deleteRule,
   System.Data.Rule updateRule
);
```

Public Properties

```
public System.Data.AcceptRejectRule get_AcceptRejectRule();
public void set_AcceptRejectRule(
   System.Data.AcceptRejectRule value
);
public System.Data.DataColumn[] get_Columns();
public System.Data.Rule get_DeleteRule();
public void set_DeleteRule(
   System.Data.Rule value
);
public System.Data.DataColumn[] get_RelatedColumns();
public System.Data.DataTable get_RelatedTable();
public System.Data.DataTable get_Table();
public System.Data.Rule get_UpdateRule();
```

```
public void set_UpdateRule(
   System.Data.Rule value
);
```

Public Methods

```
public boolean Equals(
   System.Object key
);
public int GetHashCode();
```

IColumnMapping Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IColumnMapping
```

Public Properties

```
System.String get_DataSetColumn();
void set_DataSetColumn(
   System.String value
);
System.String get_SourceColumn();
void set_SourceColumn(
   System.String value
);
```

IColumnMappingCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public interface IColumnMappingCollection
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Properties

```
System.Object get_Item(
   System.String index
);
void set_Item(
   System.String index,
   System.Object value
);
```

Public Methods

```
System.Data.IColumnMapping Add(
   System.String sourceColumnName,
   System.String dataSetColumnName
);
boolean Contains(
   System.String sourceColumnName
);
System.Data.IColumnMapping GetByDataSetColumn(
   System.String dataSetColumnName
);
int IndexOf(
   System.String sourceColumnName
);
void RemoveAt(
   System.String sourceColumnName
);
```

IDataAdapter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDataAdapter
```

Public Properties

```
System.Data.MissingMappingAction get_MissingMappingAction();
void set_MissingMappingAction(
   System.Data.MissingMappingAction value
);
System.Data.MissingSchemaAction get_MissingSchemaAction();
void set_MissingSchemaAction(
   System.Data.MissingSchemaAction value
);
System.Data.ITableMappingCollection get_TableMappings();
```

Public Methods

```
int Fill(
   System.Data.DataSet dataSet
);
System.Data.DataTable[] FillSchema(
   System.Data.DataSet dataSet,
   System.Data.SchemaType schemaType
);
System.Data.IDataParameter[] GetFillParameters();
int Update(
   System.Data.DataSet dataSet
);
```

IDataParameter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDataParameter
```

Public Properties

```
System.Data.DbType get_DbType();
void set_DbType(
   System.Data.DbType value
);
System.Data.ParameterDirection get_Direction();
void set_Direction(
   System.Data.ParameterDirection value
);
boolean get_IsNullable();
System.String get_ParameterName();
void set_ParameterName(
   System.String value
);
System.String get_SourceColumn();
void set_SourceColumn(
   System.String value
);
System.Data.DataRowVersion get_SourceVersion();
void set_SourceVersion(
   System.Data.DataRowVersion value
);
System.Object get_Value();
void set_Value(
   System.Object value
);
```

IDataParameterCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public interface IDataParameterCollection
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Properties

```
System.Object get_Item(
   System.String parameterName
);
void set_Item(
   System.String parameterName,
   System.Object value
);
```

Public Methods

```
boolean Contains(
   System.String parameterName
);
int IndexOf(
   System.String parameterName
);
void RemoveAt(
   System.String parameterName
);
```

IDataReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDataReader
   implements System.IDisposable, System.Data.IDataRecord
```

Public Properties

```
int get_Depth();
boolean get_IsClosed();
int get_RecordsAffected();
```

Public Methods

```
void Close();
System.Data.DataTable GetSchemaTable();
boolean NextResult();
boolean Read();
```

IDataRecord Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public interface IDataRecord
```

Public Properties

```
int get_FieldCount();
System.Object get_Item(
   int i
);
System.Object get_Item(
   System.String name
);
```

Public Methods

```
boolean GetBoolean(
   int i
);
ubyte GetByte(
   int i
);
long GetBytes(
   int i,
   long fieldOffset,
   ubyte[] buffer,
   int bufferoffset,
   int length
);
char GetChar(
   int i
);
long GetChars(
   int i,
   long fieldoffset,
   char[] buffer,
   int bufferoffset,
   int length
);
System.Data.IDataReader GetData(
   int i
);
System.String GetDataTypeName(
   int i
);
System.DateTime GetDateTime(
   int i
);
System.Decimal GetDecimal(
   int i
);
double GetDouble(
   int i
);
System.Type GetFieldType(
   int i
```

```
);
float GetFloat(
   int i
);
System.Guid GetGuid(
   int i
);
short GetInt16(
   int i
);
int GetInt32(
   int i
);
long GetInt64(
   int i
);
System.String GetName(
   int i
);
int GetOrdinal(
   System.String name
);
System.String GetString(
   int i
);
System.Object GetValue(
   int i
);
int GetValues(
   System.Object[] values
);
boolean IsDBNull(
   int i
);
```

IDbCommand Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDbCommand
   implements System.IDisposable
```

Public Properties

```
System.String get_CommandText();
void set_CommandText(
   System.String value
);
int get_CommandTimeout();
void set_CommandTimeout(
   int value
);
System.Data.CommandType get_CommandType();
void set_CommandType(
   System.Data.CommandType value
);
System.Data.IDbConnection get_Connection();
void set_Connection(
   System.Data.IDbConnection value
);
System.Data.IDataParameterCollection get_Parameters();
System.Data.IDbTransaction get_Transaction();
void set_Transaction(
   System.Data.IDbTransaction value
);
System.Data.UpdateRowSource get_UpdatedRowSource();
void set_UpdatedRowSource(
   System.Data.UpdateRowSource value
);
```

Public Methods

```
void Cancel();
System.Data.IDbDataParameter CreateParameter();
int ExecuteNonQuery();
System.Data.IDataReader ExecuteReader();
System.Data.IDataReader ExecuteReader(
   System.Data.CommandBehavior behavior
);
System.Object ExecuteScalar();
void Prepare();
```

IDbConnection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDbConnection
   implements System.IDisposable
```

Public Properties

```
System.String get_ConnectionString();
void set_ConnectionString(
   System.String value
);
int get_ConnectionTimeout();
System.String get_Database();
System.Data.ConnectionState get_State();
```

Public Methods

```
System.Data.IDbTransaction BeginTransaction();
System.Data.IDbTransaction BeginTransaction(
   System.Data.IsolationLevel il
);
void ChangeDatabase(
   System.String databaseName
);
void Close();
System.Data.IDbCommand CreateCommand();
void Open();
```

IDbDataAdapter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDbDataAdapter
   implements System.Data.IDataAdapter
```

Public Properties

```
System.Data.IDbCommand get_DeleteCommand();
void set_DeleteCommand(
   System.Data.IDbCommand value
);
System.Data.IDbCommand get_InsertCommand();
void set_InsertCommand(
   System.Data.IDbCommand value
);
System.Data.IDbCommand get_SelectCommand();
void set_SelectCommand(
   System.Data.IDbCommand value
);
System.Data.IDbCommand get_UpdateCommand();
void set_UpdateCommand(
   System.Data.IDbCommand value
);
```

IDbDataParameter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDbDataParameter
   implements System.Data.IDataParameter
```

Public Properties

```
ubyte get_Precision();
void set_Precision(
   ubyte value
);
ubyte get_Scale();
void set_Scale(
   ubyte value
);
int get_Size();
void set_Size(
   int value
);
```

IDbTransaction Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDbTransaction
   implements System.IDisposable
```

Public Properties

```
System.Data.IDbConnection get_Connection();
System.Data.IsolationLevel get_IsolationLevel();
```

Public Methods

```
void Commit();
void Rollback();
```

InRowChangingEventException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class InRowChangingEventException
   extends System.Data.DataException
```

Public Constructors

```
public InRowChangingEventException();
public InRowChangingEventException(
   System.String s
);
```

Protected Constructors

```
protected InRowChangingEventException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

InternalDataCollectionBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InternalDataCollectionBase
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public InternalDataCollectionBase();
```

Public Properties

```
public int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public final System.Object get_SyncRoot();
```

Public Methods

```
public final void CopyTo(
   System.Array ar,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
```

Protected Properties

```
protected System.Collections.ArrayList get_List();
```

InvalidConstraintException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class InvalidConstraintException
   extends System.Data.DataException
```

Public Constructors

```
public InvalidConstraintException();
public InvalidConstraintException(
   System.String s
);
```

Protected Constructors

```
protected InvalidConstraintException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

InvalidExpressionException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class InvalidExpressionException
   extends System.Data.DataException
```

Public Constructors

```
public InvalidExpressionException();
public InvalidExpressionException(
   System.String s
);
```

Protected Constructors

```
protected InvalidExpressionException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

IsolationLevel Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum IsolationLevel
```

Members

Chaos

ReadCommitted

ReadUncommitted

RepeatableRead

Serializable

Unspecified

ITableMapping Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ITableMapping
```

Public Properties

```
System.Data.IColumnMappingCollection get_ColumnMappings();
System.String get_DataSetTable();
void set_DataSetTable(
   System.String value
);
System.String get_SourceTable();
void set_SourceTable(
   System.String value
);
```

ITableMappingCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public interface ITableMappingCollection
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Properties

```
System.Object get_Item(
   System.String index
);
void set_Item(
   System.String index,
   System.Object value
);
```

Public Methods

```
System.Data.ITableMapping Add(
   System.String sourceTableName,
   System.String dataSetTableName
);
boolean Contains(
   System.String sourceTableName
);
System.Data.ITableMapping GetByDataSetTable(
   System.String dataSetTableName
);
int IndexOf(
   System.String sourceTableName
);
void RemoveAt(
   System.String sourceTableName
);
```

MappingType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MappingType
```

Members

Attribute

Element

Hidden

SimpleContent

MergeFailedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MergeFailedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public MergeFailedEventArgs(
   System.Data.DataTable table,
   System.String conflict
);
```

Public Properties

```
public System.String get_Conflict();
public System.Data.DataTable get_Table();
```

MergeFailedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void MergeFailedEventHandler(
   System.Object sender,
   System.Data.MergeFailedEventArgs e
);
```

MissingMappingAction Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MissingMappingAction
```

Members

Error

Ignore

Passthrough

MissingPrimaryKeyException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class MissingPrimaryKeyException
   extends System.Data.DataException
```

Public Constructors

```
public MissingPrimaryKeyException();
public MissingPrimaryKeyException(
   System.String s
);
```

Protected Constructors

```
protected MissingPrimaryKeyException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

MissingSchemaAction Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MissingSchemaAction
```

Members

Add

AddWithKey

Error

Ignore

NoNullAllowedException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class NoNullAllowedException
   extends System.Data.DataException
```

Public Constructors

```
public NoNullAllowedException();
public NoNullAllowedException(
   System.String s
);
```

Protected Constructors

```
protected NoNullAllowedException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ParameterDirection Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ParameterDirection
```

Members

Input

InputOutput

Output

ReturnValue

PropertyAttributes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum PropertyAttributes
```

Members

NotSupported

Optional

Read

Required

Write

PropertyCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PropertyCollection
   extends System.Collections.Hashtable
```

Public Constructors

```
public PropertyCollection();
```

ReadOnlyException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ReadOnlyException
   extends System.Data.DataException
```

Public Constructors

```
public ReadOnlyException();
public ReadOnlyException(
   System.String s
);
```

Protected Constructors

```
protected ReadOnlyException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

RowNotInTableException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RowNotInTableException
   extends System.Data.DataException
```

Public Constructors

```
public RowNotInTableException();
public RowNotInTableException(
   System.String s
);
```

Protected Constructors

```
protected RowNotInTableException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Rule Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum Rule
```

Members

Cascade

None

SetDefault

SetNull

SchemaType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SchemaType
```

Members

Mapped

Source

SqlDbType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SqlDbType
```

Members

BigInt

Binary

Bit

Char

DateTime

Decimal

Float

Image

Int

Money

NChar

NText

NVarChar

Real

SmallDateTime

SmallInt

SmallMoney

Text

Timestamp

TinyInt

UniqueIdentifier

VarBinary

VarChar

Variant

StateChangeEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class StateChangeEventArgs
   extends System.EventArgs
```

Public Constructors

```
public StateChangeEventArgs(
   System.Data.ConnectionState originalState,
   System.Data.ConnectionState currentState
);
```

Public Properties

```
public System.Data.ConnectionState get_CurrentState();
public System.Data.ConnectionState get_OriginalState();
```

StateChangeEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void StateChangeEventHandler(
   System.Object sender,
   System.Data.StateChangeEventArgs e
);
```

StatementType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum StatementType
```

Members

Delete

Insert

Select

Update

StrongTypingException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StrongTypingException
   extends System.Data.DataException
```

Public Constructors

```
public StrongTypingException();
public StrongTypingException(
   System.String s,
   System.Exception innerException
);
```

Protected Constructors

```
protected StrongTypingException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

SyntaxErrorException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SyntaxErrorException
   extends System.Data.InvalidExpressionException
```

Public Constructors

```
public SyntaxErrorException();
public SyntaxErrorException(
   System.String s
);
```

Protected Constructors

```
protected SyntaxErrorException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

TypedDataSetGenerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TypedDataSetGenerator
   extends System.Object
```

Public Constructors

```
public TypedDataSetGenerator();
```

Public Methods

```
public static void Generate(
   System.Data.DataSet dataSet,
   System.CodeDom.CodeNamespace codeNamespace,
   System.CodeDom.Compiler.ICodeGenerator codeGen
);
public static System.String GenerateIdName(
   System.String name,
   System.CodeDom.Compiler.ICodeGenerator codeGen
);
```

TypedDataSetGeneratorException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class TypedDataSetGeneratorException
   extends System.Data.DataException
```

Public Constructors

```
public TypedDataSetGeneratorException();
public TypedDataSetGeneratorException(
   System.Collections.ArrayList list
);
```

Public Properties

```
public System.Collections.ArrayList get_ErrorList();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected TypedDataSetGeneratorException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

UniqueConstraint Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.UniqueConstraintEditor, Micros
oft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "ConstraintName") */
public class UniqueConstraint
   extends System.Data.Constraint
```

Public Constructors

```
public UniqueConstraint(
   System.Data.DataColumn column
);
public UniqueConstraint(
   System.Data.DataColumn[] columns
);
public UniqueConstraint(
   System.Data.DataColumn column,
   boolean isPrimaryKey
);
public UniqueConstraint(
   System.Data.DataColumn[] columns,
   boolean isPrimaryKey
);
public UniqueConstraint(
   System.String name,
   System.Data.DataColumn column
);
public UniqueConstraint(
   System.String name,
   System.Data.DataColumn[] columns
);
public UniqueConstraint(
   System.String name,
   System.Data.DataColumn column,
   boolean isPrimaryKey
);
public UniqueConstraint(
   System.String name,
   System.Data.DataColumn[] columns,
   boolean isPrimaryKey
);
/** @attribute System.ComponentModel.BrowsableAttribute(Browsable = false) */
public UniqueConstraint(
   System.String name,
   System.String[] columnNames,
   boolean isPrimaryKey
);
```

Public Properties

```
public System.Data.DataColumn[] get_Columns();
public boolean get_IsPrimaryKey();
public System.Data.DataTable get_Table();
```

Public Methods

```
public boolean Equals(
   System.Object key2
);
public int GetHashCode();
```

UpdateRowSource Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum UpdateRowSource
```

Members

Both

FirstReturnedRecord

None

OutputParameters

UpdateStatus Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum UpdateStatus
```

Members

Continue

ErrorsOccurred

SkipAllRemainingRows

SkipCurrentRow

VersionNotFoundException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class VersionNotFoundException
   extends System.Data.DataException
```

Public Constructors

```
public VersionNotFoundException();
public VersionNotFoundException(
   System.String s
);
```

Protected Constructors

```
protected VersionNotFoundException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

XmlReadMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlReadMode
```

Members

Auto

DiffGram

Fragment

IgnoreSchema

InferSchema

ReadSchema

XmlWriteMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlWriteMode
```

Members

DiffGram

IgnoreSchema

WriteSchema

System.Data.Common Namespace (Visual J# Syntax)

Classes

DataAdapter

DataColumnMapping

DataColumnMappingCollection

DataTableMapping

DataTableMappingCollection

DbDataAdapter

DBDataPermission

DBDataPermissionAttribute

DbDataRecord

DbEnumerator

RowUpdatedEventArgs

RowUpdatingEventArgs

DataAdapter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class DataAdapter
   extends System.ComponentModel.Component
   implements System.Data.IDataAdapter
```

Public Properties

```
public boolean get_AcceptChangesDuringFill();
public void set_AcceptChangesDuringFill(
   boolean value
);
public boolean get_ContinueUpdateOnError();
public void set_ContinueUpdateOnError(
   boolean value
);
public final System.Data.MissingMappingAction get_MissingMappingAction();
public final void set_MissingMappingAction(
   System.Data.MissingMappingAction value
);
public final System.Data.MissingSchemaAction get_MissingSchemaAction();
public final void set_MissingSchemaAction(
   System.Data.MissingSchemaAction value
);
public System.Data.Common.DataTableMappingCollection get_TableMappings();
```

Public Methods

```
public abstract int Fill(
   System.Data.DataSet dataSet
);
public abstract System.Data.DataTable[] FillSchema(
   System.Data.DataSet dataSet,
   System.Data.SchemaType schemaType
);
public abstract System.Data.IDataParameter[] GetFillParameters();
public abstract int Update(
   System.Data.DataSet dataSet
);
```

Protected Constructors

```
protected DataAdapter();
```

Protected Methods

```
protected System.Data.Common.DataAdapter CloneInternals();
protected System.Data.Common.DataTableMappingCollection CreateTableMappings();
protected void Dispose(
   boolean disposing
);
protected boolean ShouldSerializeTableMappings();
```

DataColumnMapping Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Data.
Common.DataColumnMappingConverter, System.Data, Version=1.0.3300.0, Culture=neutral, PublicKe
yToken=b77a5c561934e089") */
public class DataColumnMapping
   extends System.MarshalByRefObject
   implements System.Data.IColumnMapping, System.ICloneable
```

Public Constructors

```
public DataColumnMapping();
public DataColumnMapping(
   System.String sourceColumn,
   System.String dataSetColumn
);
```

Public Properties

```
public final System.String get_DataSetColumn();
public final void set_DataSetColumn(
   System.String value
);
public final System.String get_SourceColumn();
public final void set_SourceColumn(
   System.String value
);
```

Public Methods

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public System.Data.DataColumn GetDataColumnBySchemaAction(
   System.Data.DataTable dataTable,
   System.Type dataType,
   System.Data.MissingSchemaAction schemaAction
);
public System.String ToString();
```

DataColumnMappingCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DataColumnMappingCollection
   extends System.MarshalByRefObject
   implements System.Data.IColumnMappingCollection, System.Collections.IList, System.Collecti
ons.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public DataColumnMappingCollection();
```

Public Properties

```
public final int get_Count();
public System.Data.Common.DataColumnMapping get_Item(
   int index
);
public System.Data.Common.DataColumnMapping get_Item(
   System.String sourceColumn
);
public void set_Item(
   int index,
   System.Data.Common.DataColumnMapping value
);
public void set_Item(
   System.String sourceColumn,
   System.Data.Common.DataColumnMapping value
);
```

Public Methods

```
public final int Add(
   System.Object value
);
public System.Data.Common.DataColumnMapping Add(
   System.String sourceColumn,
   System.String dataSetColumn
);
public void AddRange(
   System.Data.Common.DataColumnMapping[] values
);
public final void Clear();
public final boolean Contains(
   System.Object value
);
public final boolean Contains(
   System.String value
);
public final void CopyTo(
   System.Array array,
   int index
);
public System.Data.Common.DataColumnMapping GetByDataSetColumn(
   System.String value
```

```
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public static System.Data.Common.DataColumnMapping GetColumnMappingBySchemaAction(
   System.Data.Common.DataColumnMappingCollection columnMappings,
   System.String sourceColumn,
   System.Data.MissingMappingAction mappingAction
);
public final System.Collections.IEnumerator GetEnumerator();
public final int IndexOf(
   System.Object value
);
public final int IndexOf(
   System.String sourceColumn
);
public int IndexOfDataSetColumn(
   System.String dataSetColumn
);
public final void Insert(
   int index,
   System.Object value
);
public final void Remove(
   System.Object value
);
public final void RemoveAt(
   int index
);
public final void RemoveAt(
   System.String sourceColumn
);
```

DataTableMapping Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Data.
Common.DataTableMappingConverter, System.Data, Version=1.0.3300.0, Culture=neutral, PublicKey
Token=b77a5c561934e089") */
public class DataTableMapping
   extends System.MarshalByRefObject
   implements System.Data.ITableMapping, System.ICloneable
```

Public Constructors

```
public DataTableMapping();
public DataTableMapping(
   System.String sourceTable,
   System.String dataSetTable
);
public DataTableMapping(
   System.String sourceTable,
   System.String dataSetTable,
   System.Data.Common.DataColumnMapping[] columnMappings
);
```

Public Properties

```
public System.Data.Common.DataColumnMappingCollection get_ColumnMappings();
public final System.String get_DataSetTable();
public final void set_DataSetTable(
   System.String value
);
public final System.String get_SourceTable();
public final void set_SourceTable(
   System.String value
);
```

Public Methods

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public System.Data.Common.DataColumnMapping GetColumnMappingBySchemaAction(
   System.String sourceColumn,
   System.Data.MissingMappingAction mappingAction
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public System.Data.DataTable GetDataTableBySchemaAction(
   System.Data.DataSet dataSet,
   System.Data.MissingSchemaAction schemaAction
);
public System.String ToString();
```

DataTableMappingCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.DataTableMappingCollectionEdit
or, Microsoft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3
a") */
/** @attribute System.ComponentModel.ListBindableAttribute(ListBindable = false) */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DataTableMappingCollection
   extends System.MarshalByRefObject
   implements System.Data.ITableMappingCollection, System.Collections.IList, System.Collectio
ns.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public DataTableMappingCollection();
```

Public Properties

```
public final int get_Count();
public System.Data.Common.DataTableMapping get_Item(
   int index
);
public System.Data.Common.DataTableMapping get_Item(
   System.String sourceTable
);
public void set_Item(
   int index,
   System.Data.Common.DataTableMapping value
);
public void set_Item(
   System.String sourceTable,
   System.Data.Common.DataTableMapping value
);
```

Public Methods

```
public final int Add(
   System.Object value
);
public System.Data.Common.DataTableMapping Add(
   System.String sourceTable,
   System.String dataSetTable
);
public void AddRange(
   System.Data.Common.DataTableMapping[] values
);
public final void Clear();
public final boolean Contains(
   System.Object value
);
public final boolean Contains(
   System.String value
);
```

```
public final void CopyTo(
   System.Array array,
   int index
);
public System.Data.Common.DataTableMapping GetByDataSetTable(
   System.String dataSetTable
);
public final System.Collections.IEnumerator GetEnumerator();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public static System.Data.Common.DataTableMapping GetTableMappingBySchemaAction(
   System.Data.Common.DataTableMappingCollection tableMappings,
   System.String sourceTable,
   System.String dataSetTable,
   System.Data.MissingMappingAction mappingAction
);
public final int IndexOf(
   System.Object value
);
public final int IndexOf(
   System.String sourceTable
);
public int IndexOfDataSetTable(
   System.String dataSetTable
);
public final void Insert(
   int index,
   System.Object value
);
public final void Remove(
   System.Object value
);
public final void RemoveAt(
   int index
);
public final void RemoveAt(
   System.String sourceTable
);
```

DbDataAdapter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class DbDataAdapter
   extends System.Data.Common.DataAdapter
   implements System.ICloneable
```

Public Methods

```
public int Fill(
   System.Data.DataSet dataSet
);
public int Fill(
   System.Data.DataTable dataTable
);
public int Fill(
   System.Data.DataSet dataSet,
   System.String srcTable
);
public int Fill(
   System.Data.DataSet dataSet,
   int startRecord,
   int maxRecords,
   System.String srcTable
);
public System.Data.DataTable FillSchema(
   System.Data.DataTable dataTable,
   System.Data.SchemaType schemaType
);
public System.Data.DataTable[] FillSchema(
   System.Data.DataSet dataSet,
   System.Data.SchemaType schemaType
);
public System.Data.DataTable[] FillSchema(
   System.Data.DataSet dataSet,
   System.Data.SchemaType schemaType,
   System.String srcTable
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public System.Data.IDataParameter[] GetFillParameters();
public int Update(
   System.Data.DataRow[] dataRows
);
public int Update(
   System.Data.DataSet dataSet
);
public int Update(
   System.Data.DataTable dataTable
);
public int Update(
   System.Data.DataSet dataSet,
   System.String srcTable
);
```

Public Events

```
public void add_FillError(
   System.Data.FillErrorEventHandler value
```

```
);
public void remove_FillError(
   System.Data.FillErrorEventHandler value
);
```

Public Fields

```
public static System.String DefaultSourceTableName;
```

Protected Constructors

```
protected DbDataAdapter();
```

Protected Methods

```
protected abstract System.Data.Common.RowUpdatedEventArgs CreateRowUpdatedEvent(
   System.Data.DataRow dataRow,
   System.Data.IDbCommand command,
   System.Data.StatementType statementType,
   System.Data.Common.DataTableMapping tableMapping
);
protected abstract System.Data.Common.RowUpdatingEventArgs CreateRowUpdatingEvent(
   System.Data.DataRow dataRow,
   System.Data.IDbCommand command,
   System.Data.StatementType statementType,
   System.Data.Common.DataTableMapping tableMapping
);
protected void Dispose(
   boolean disposing
);
protected int Fill(
   System.Data.DataTable dataTable,
   System.Data.IDataReader dataReader
);
protected int Fill(
   System.Data.DataTable dataTable,
   System.Data.IDbCommand command,
   System.Data.CommandBehavior behavior
);
protected int Fill(
   System.Data.DataSet dataSet,
   System.String srcTable,
   System.Data.IDataReader dataReader,
   int startRecord,
   int maxRecords
);
protected int Fill(
   System.Data.DataSet dataSet,
   int startRecord,
   int maxRecords,
   System.String srcTable,
   System.Data.IDbCommand command,
   System.Data.CommandBehavior behavior
);
protected System.Data.DataTable FillSchema(
   System.Data.DataTable dataTable,
   System.Data.SchemaType schemaType,
   System.Data.IDbCommand command,
   System.Data.CommandBehavior behavior
);
protected System.Data.DataTable[] FillSchema(
   System.Data.DataSet dataSet,
```

```
    System.Data.SchemaType schemaType,
    System.Data.IDbCommand command,
    System.String srcTable,
    System.Data.CommandBehavior behavior
  );
  protected void OnFillError(
    System.Data.FillErrorEventArgs value
  );
  protected abstract void OnRowUpdated(
    System.Data.Common.RowUpdatedEventArgs value
  );
  protected abstract void OnRowUpdating(
    System.Data.Common.RowUpdatingEventArgs value
  );
  protected int Update(
    System.Data.DataRow[] dataRows,
    System.Data.Common.DataTableMapping tableMapping
  );
```

DBDataPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class DBDataPermission
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Constructors

```
public DBDataPermission(
   System.Security.Permissions.PermissionState state,
   boolean allowBlankPassword
);
```

Public Properties

```
public boolean get_AllowBlankPassword();
public void set_AllowBlankPassword(
   boolean value
);
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement securityElement
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public final boolean IsUnrestricted();
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

Protected Constructors

```
protected DBDataPermission();
protected DBDataPermission(
   System.Security.Permissions.PermissionState state
);
```

Protected Methods

```
protected System.Data.Common.DBDataPermission CreateInstance();
```

DBDataPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public abstract class DBDataPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Properties

```
public boolean get_AllowBlankPassword();
public void set_AllowBlankPassword(
   boolean value
);
```

Protected Constructors

```
protected DBDataPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

DbDataRecord Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DbDataRecord
   extends System.Object
   implements System.Data.IDataRecord, System.ComponentModel.ICustomTypeDescriptor
```

Public Properties

```
public final int get_FieldCount();
public final System.Object get_Item(
   int i
);
public final System.Object get_Item(
   System.String name
);
```

Public Methods

```
public final boolean GetBoolean(
   int i
);
public final ubyte GetByte(
   int i
);
public final long GetBytes(
   int i,
   long dataIndex,
   ubyte[] buffer,
   int bufferIndex,
   int length
);
public final char GetChar(
   int i
);
public final long GetChars(
   int i,
   long dataIndex,
   char[] buffer,
   int bufferIndex,
   int length
);
public final System.Data.IDataReader GetData(
   int i
);
public final System.String GetDataTypeName(
   int i
);
public final System.DateTime GetDateTime(
   int i
);
public final System.Decimal GetDecimal(
   int i
);
public final double GetDouble(
   int i
);
```

```
public final System.Type GetFieldType(
   int i
);
public final float GetFloat(
   int i
);
public final System.Guid GetGuid(
   int i
);
public final short GetInt16(
   int i
);
public final int GetInt32(
   int i
);
public final long GetInt64(
   int i
);
public final System.String GetName(
   int i
);
public final int GetOrdinal(
   System.String name
);
public final System.String GetString(
   int i
);
public final System.Object GetValue(
   int i
);
public final int GetValues(
   System.Object[] values
);
public final boolean IsDBNull(
   int i
);
```

DbEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DbEnumerator
   extends System.Object
   implements System.Collections.IEnumerator
```

Public Constructors

```
public DbEnumerator(
   System.Data.IDataReader reader
);
public DbEnumerator(
   System.Data.IDataReader reader,
   boolean closeReader
);
```

Public Properties

```
public final System.Object get_Current();
```

Public Methods

```
public final boolean MoveNext();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public final void Reset();
```

RowUpdatedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class RowUpdatedEventArgs
   extends System.EventArgs
```

Public Properties

```
public System.Data.IDbCommand get_Command();
public System.Exception get_Errors();
public void set_Errors(
   System.Exception value
);
public int get_RecordsAffected();
public System.Data.DataRow get_Row();
public System.Data.StatementType get_StatementType();
public System.Data.UpdateStatus get_Status();
public void set_Status(
   System.Data.UpdateStatus value
);
public System.Data.Common.DataTableMapping get_TableMapping();
```

Protected Constructors

```
protected RowUpdatedEventArgs(
   System.Data.DataRow dataRow,
   System.Data.IDbCommand command,
   System.Data.StatementType statementType,
   System.Data.Common.DataTableMapping tableMapping
);
```

RowUpdatingEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class RowUpdatingEventArgs
   extends System.EventArgs
```

Public Properties

```
public System.Data.IDbCommand get_Command();
public void set_Command(
   System.Data.IDbCommand value
);
public System.Exception get_Errors();
public void set_Errors(
   System.Exception value
);
public System.Data.DataRow get_Row();
public System.Data.StatementType get_StatementType();
public System.Data.UpdateStatus get_Status();
public void set_Status(
   System.Data.UpdateStatus value
);
public System.Data.Common.DataTableMapping get_TableMapping();
```

Protected Constructors

```
protected RowUpdatingEventArgs(
   System.Data.DataRow dataRow,
   System.Data.IDbCommand command,
   System.Data.StatementType statementType,
   System.Data.Common.DataTableMapping tableMapping
);
```

System.Data.OleDb Namespace (Visual J# Syntax)

Classes

OleDbCommand

OleDbCommandBuilder

OleDbConnection

OleDbDataAdapter

OleDbDataReader

OleDbError

OleDbErrorCollection

OleDbException

OleDbInfoMessageEventArgs

OleDbParameter

OleDbParameterCollection

OleDbPermission

OleDbPermissionAttribute

OleDbRowUpdatedEventArgs

OleDbRowUpdatingEventArgs

OleDbSchemaGuid

OleDbTransaction

Delegates

OleDbInfoMessageEventHandler

OleDbRowUpdatedEventHandler

OleDbRowUpdatingEventHandler

Enumerations

OleDbLiteral

OleDbType

OleDbCommand Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "Microsoft.VSDesigner.Data.VS.OleDbCommandDesig
ner, Microsoft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a
3a") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemType = System.Drawing.De
sign.ToolboxItem, ToolboxItemTypeName = "System.Drawing.Design.ToolboxItem, System.Drawing, V
ersion=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class OleDbCommand
   extends System.ComponentModel.Component
   implements System.ICloneable, System.Data.IDbCommand
```

Public Constructors

```
public OleDbCommand();
public OleDbCommand(
   System.String cmdText
);
public OleDbCommand(
   System.String cmdText,
   System.Data.OleDb.OleDbConnection connection
);
public OleDbCommand(
   System.String cmdText,
   System.Data.OleDb.OleDbConnection connection,
   System.Data.OleDb.OleDbTransaction transaction
);
```

Public Properties

```
public final System.String get_CommandText();
public final void set_CommandText(
   System.String value
);
public final int get_CommandTimeout();
public final void set_CommandTimeout(
   int value
);
public final System.Data.CommandType get_CommandType();
public final void set_CommandType(
   System.Data.CommandType value
);
public System.Data.OleDb.OleDbConnection get_Connection();
public void set_Connection(
   System.Data.OleDb.OleDbConnection value
);
public boolean get_DesignTimeVisible();
public void set_DesignTimeVisible(
   boolean value
);
public System.Data.OleDb.OleDbParameterCollection get_Parameters();
public System.Data.OleDb.OleDbTransaction get_Transaction();
public void set_Transaction(
   System.Data.OleDb.OleDbTransaction value
);
public final System.Data.UpdateRowSource get_UpdatedRowSource();
```

```
public final void set_UpdatedRowSource(
   System.Data.UpdateRowSource value
);
```

Public Methods

```
public final void Cancel();
public System.Data.OleDb.OleDbParameter CreateParameter();
public final int ExecuteNonQuery();
public System.Data.OleDb.OleDbDataReader ExecuteReader();
public System.Data.OleDb.OleDbDataReader ExecuteReader(
   System.Data.CommandBehavior behavior
);
public final System.Object ExecuteScalar();
public final void Prepare();
public void ResetCommandTimeout();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

OleDbCommandBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class OleDbCommandBuilder
   extends System.ComponentModel.Component
```

Public Constructors

```
public OleDbCommandBuilder();
public OleDbCommandBuilder(
   System.Data.OleDb.OleDbDataAdapter adapter
);
```

Public Properties

```
public System.Data.OleDb.OleDbDataAdapter get_DataAdapter();
public void set_DataAdapter(
   System.Data.OleDb.OleDbDataAdapter value
);
public System.String get_QuotePrefix();
public void set_QuotePrefix(
   System.String value
);
public System.String get_QuoteSuffix();
public void set_QuoteSuffix(
   System.String value
);
```

Public Methods

```
public static void DeriveParameters(
   System.Data.OleDb.OleDbCommand command
);
public System.Data.OleDb.OleDbCommand GetDeleteCommand();
public System.Data.OleDb.OleDbCommand GetInsertCommand();
public System.Data.OleDb.OleDbCommand GetUpdateCommand();
public void RefreshSchema();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

OleDbConnection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "InfoMessage") */
public class OleDbConnection
   extends System.ComponentModel.Component
   implements System.ICloneable, System.Data.IDbConnection
```

Public Constructors

```
public OleDbConnection();
public OleDbConnection(
   System.String connectionString
);
```

Public Properties

```
public final System.String get_ConnectionString();
public final void set_ConnectionString(
   System.String value
);
public final int get_ConnectionTimeout();
public final System.String get_Database();
public System.String get_DataSource();
public System.String get_Provider();
public System.String get_ServerVersion();
public final System.Data.ConnectionState get_State();
```

Public Methods

```
public System.Data.OleDb.OleDbTransaction BeginTransaction();
public System.Data.OleDb.OleDbTransaction BeginTransaction(
   System.Data.IsolationLevel isolationLevel
);
public final void ChangeDatabase(
   System.String value
);
public final void Close();
public System.Data.OleDb.OleDbCommand CreateCommand();
public System.Data.DataTable GetOleDbSchemaTable(
   System.Guid schema,
   System.Object[] restrictions
);
public final void Open();
public static void ReleaseObjectPool();
```

Public Events

```
public void add_InfoMessage(
   System.Data.OleDb.OleDbInfoMessageEventHandler value
);
public void remove_InfoMessage(
   System.Data.OleDb.OleDbInfoMessageEventHandler value
);
```

```
public void add_StateChange(
   System.Data.StateChangeEventHandler value
);
public void remove_StateChange(
   System.Data.StateChangeEventHandler value
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

OleDbDataAdapter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "Microsoft.VSDesigner.Data.VS.OleDbDataAdapterD
esigner, Microsoft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11
d50a3a") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "Microsoft.VS
Designer.Data.VS.OleDbDataAdapterToolboxItem, Microsoft.VSDesigner, Version=7.0.3300.0, Cultu
re=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "RowUpdated") */
public class OleDbDataAdapter
   extends System.Data.Common.DbDataAdapter
   implements System.Data.IDbDataAdapter
```

Public Constructors

```
public OleDbDataAdapter();
public OleDbDataAdapter(
   System.Data.OleDb.OleDbCommand selectCommand
);
public OleDbDataAdapter(
   System.String selectCommandText,
   System.Data.OleDb.OleDbConnection selectConnection
);
public OleDbDataAdapter(
   System.String selectCommandText,
   System.String selectConnectionString
);
```

Public Properties

```
public System.Data.OleDb.OleDbCommand get_DeleteCommand();
public void set_DeleteCommand(
   System.Data.OleDb.OleDbCommand value
);
public System.Data.OleDb.OleDbCommand get_InsertCommand();
public void set_InsertCommand(
   System.Data.OleDb.OleDbCommand value
);
public System.Data.OleDb.OleDbCommand get_SelectCommand();
public void set_SelectCommand(
   System.Data.OleDb.OleDbCommand value
);
public System.Data.OleDb.OleDbCommand get_UpdateCommand();
public void set_UpdateCommand(
   System.Data.OleDb.OleDbCommand value
);
```

Public Methods

```
public int Fill(
   System.Data.DataTable dataTable,
   System.Object ADODBRecordSet
);
public int Fill(
```

```
   System.Data.DataSet dataSet,
   System.Object ADODBRecordSet,
   System.String srcTable
);
```

Public Events

```
public void add_RowUpdated(
   System.Data.OleDb.OleDbRowUpdatedEventHandler value
);
public void remove_RowUpdated(
   System.Data.OleDb.OleDbRowUpdatedEventHandler value
);
public void add_RowUpdating(
   System.Data.OleDb.OleDbRowUpdatingEventHandler value
);
public void remove_RowUpdating(
   System.Data.OleDb.OleDbRowUpdatingEventHandler value
);
```

Protected Methods

```
protected System.Data.Common.RowUpdatedEventArgs CreateRowUpdatedEvent(
   System.Data.DataRow dataRow,
   System.Data.IDbCommand command,
   System.Data.StatementType statementType,
   System.Data.Common.DataTableMapping tableMapping
);
protected System.Data.Common.RowUpdatingEventArgs CreateRowUpdatingEvent(
   System.Data.DataRow dataRow,
   System.Data.IDbCommand command,
   System.Data.StatementType statementType,
   System.Data.Common.DataTableMapping tableMapping
);
protected void OnRowUpdated(
   System.Data.Common.RowUpdatedEventArgs value
);
protected void OnRowUpdating(
   System.Data.Common.RowUpdatingEventArgs value
);
```

OleDbDataReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class OleDbDataReader
   extends System.MarshalByRefObject
   implements System.Data.IDataReader, System.IDisposable, System.Data.IDataRecord, System.Co
llections.IEnumerable
```

Public Properties

```
public final int get_Depth();
public final int get_FieldCount();
public final boolean get_IsClosed();
public final System.Object get_Item(
   int index
);
public final System.Object get_Item(
   System.String name
);
public final int get_RecordsAffected();
```

Public Methods

```
public final void Close();
public final boolean GetBoolean(
   int ordinal
);
public final ubyte GetByte(
   int ordinal
);
public final long GetBytes(
   int ordinal,
   long dataIndex,
   ubyte[] buffer,
   int bufferIndex,
   int length
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public final char GetChar(
   int ordinal
);
public final long GetChars(
   int ordinal,
   long dataIndex,
   char[] buffer,
   int bufferIndex,
   int length
);
public System.Data.OleDb.OleDbDataReader GetData(
   int ordinal
);
public final System.String GetDataTypeName(
   int index
);
public final System.DateTime GetDateTime(
   int ordinal
);
```

```
public final System.Decimal GetDecimal(
   int ordinal
);
public final double GetDouble(
   int ordinal
);
public final System.Type GetFieldType(
   int index
);
public final float GetFloat(
   int ordinal
);
public final System.Guid GetGuid(
   int ordinal
);
public final short GetInt16(
   int ordinal
);
public final int GetInt32(
   int ordinal
);
public final long GetInt64(
   int ordinal
);
public final System.String GetName(
   int index
);
public final int GetOrdinal(
   System.String name
);
public final System.Data.DataTable GetSchemaTable();
public final System.String GetString(
   int ordinal
);
public System.TimeSpan GetTimeSpan(
   int ordinal
);
public final System.Object GetValue(
   int ordinal
);
public final int GetValues(
   System.Object[] values
);
public final boolean IsDBNull(
   int ordinal
);
public final boolean NextResult();
public final boolean Read();
```

Protected Methods

```
protected void Finalize();
```

OleDbError Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class OleDbError
   extends System.Object
```

Public Properties

```
public System.String get_Message();
public int get_NativeError();
public System.String get_Source();
public System.String get_SQLState();
```

Public Methods

```
public System.String ToString();
```

OleDbErrorCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.ListBindableAttribute(ListBindable = false) */
public class OleDbErrorCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public System.Data.OleDb.OleDbError get_Item(
   int index
);
```

Public Methods

```
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
```

OleDbException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class OleDbException
   extends System.Runtime.InteropServices.ExternalException
```

Public Properties

```
public int get_ErrorCode();
public System.Data.OleDb.OleDbErrorCollection get_Errors();
public System.String get_Message();
public System.String get_Source();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo si,
   System.Runtime.Serialization.StreamingContext context
);
```

OleDbInfoMessageEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class OleDbInfoMessageEventArgs
   extends System.EventArgs
```

Public Properties

```
public int get_ErrorCode();
public System.Data.OleDb.OleDbErrorCollection get_Errors();
public System.String get_Message();
public System.String get_Source();
```

Public Methods

```
public System.String ToString();
```

OleDbInfoMessageEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void OleDbInfoMessageEventHandler(
   System.Object sender,
   System.Data.OleDb.OleDbInfoMessageEventArgs e
);
```

OleDbLiteral Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum OleDbLiteral
```

Members

Binary_Literal

Catalog_Name

Catalog_Separator

Char_Literal

Column_Alias

Column_Name

Correlation_Name

Cube_Name

Cursor_Name

Dimension_Name

Escape_Percent_Prefix

Escape_Percent_Suffix

Escape_Underscore_Prefix

Escape_Underscore_Suffix

Hierarchy_Name

Index_Name

Invalid

Level_Name

Like_Percent

Like_Underscore

Member_Name

Procedure_Name

Property_Name

Quote_Prefix

Quote_Suffix

Schema_Name

Schema_Separator

Table_Name

Text_Command

User_Name

View_Name

OleDbParameter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Data.
OleDb.OleDbParameterConverter, System.Data, Version=1.0.3300.0, Culture=neutral, PublicKeyTok
en=b77a5c561934e089") */
public class OleDbParameter
   extends System.MarshalByRefObject
   implements System.Data.IDbDataParameter, System.Data.IDataParameter, System.ICloneable
```

Public Constructors

```
public OleDbParameter();
public OleDbParameter(
   System.String name,
   System.Data.OleDb.OleDbType dataType
);
public OleDbParameter(
   System.String name,
   System.Object value
);
public OleDbParameter(
   System.String name,
   System.Data.OleDb.OleDbType dataType,
   int size
);
public OleDbParameter(
   System.String name,
   System.Data.OleDb.OleDbType dataType,
   int size,
   System.String srcColumn
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public OleDbParameter(
   System.String parameterName,
   System.Data.OleDb.OleDbType dbType,
   int size,
   System.Data.ParameterDirection direction,
   boolean isNullable,
   ubyte precision,
   ubyte scale,
   System.String srcColumn,
   System.Data.DataRowVersion srcVersion,
   System.Object value
);
```

Public Properties

```
public final System.Data.DbType get_DbType();
public final void set_DbType(
   System.Data.DbType value
);
public final System.Data.ParameterDirection get_Direction();
public final void set_Direction(
   System.Data.ParameterDirection value
);
public final boolean get_IsNullable();
public void set_IsNullable(
```

```
      boolean value
   );
   public System.Data.OleDb.OleDbType get_OleDbType();
   public void set_OleDbType(
      System.Data.OleDb.OleDbType value
   );
   public final System.String get_ParameterName();
   public final void set_ParameterName(
      System.String value
   );
   public final ubyte get_Precision();
   public final void set_Precision(
      ubyte value
   );
   public final ubyte get_Scale();
   public final void set_Scale(
      ubyte value
   );
   public final int get_Size();
   public final void set_Size(
      int value
   );
   public final System.String get_SourceColumn();
   public final void set_SourceColumn(
      System.String value
   );
   public final System.Data.DataRowVersion get_SourceVersion();
   public final void set_SourceVersion(
      System.Data.DataRowVersion value
   );
   public final System.Object get_Value();
   public final void set_Value(
      System.Object value
   );
```

Public Methods

```
   public System.String ToString();
```

OleDbParameterCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.DBParametersEditor, Microsoft.
VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.ListBindableAttribute(ListBindable = false) */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class OleDbParameterCollection
   extends System.MarshalByRefObject
   implements System.Data.IDataParameterCollection, System.Collections.IList, System.Collecti
ons.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public System.Data.OleDb.OleDbParameter get_Item(
   int index
);
public System.Data.OleDb.OleDbParameter get_Item(
   System.String parameterName
);
public void set_Item(
   int index,
   System.Data.OleDb.OleDbParameter value
);
public void set_Item(
   System.String parameterName,
   System.Data.OleDb.OleDbParameter value
);
```

Public Methods

```
public final int Add(
   System.Object value
);
public System.Data.OleDb.OleDbParameter Add(
   System.Data.OleDb.OleDbParameter value
);
public System.Data.OleDb.OleDbParameter Add(
   System.String parameterName,
   System.Data.OleDb.OleDbType oleDbType
);
public System.Data.OleDb.OleDbParameter Add(
   System.String parameterName,
   System.Object value
);
public System.Data.OleDb.OleDbParameter Add(
   System.String parameterName,
   System.Data.OleDb.OleDbType oleDbType,
   int size
);
public System.Data.OleDb.OleDbParameter Add(
   System.String parameterName,
   System.Data.OleDb.OleDbType oleDbType,
   int size,
   System.String sourceColumn
```

```
);
public final void Clear();
public final boolean Contains(
   System.Object value
);
public final boolean Contains(
   System.String value
);
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public final int IndexOf(
   System.Object value
);
public final int IndexOf(
   System.String parameterName
);
public final void Insert(
   int index,
   System.Object value
);
public final void Remove(
   System.Object value
);
public final void RemoveAt(
   int index
);
public final void RemoveAt(
   System.String parameterName
);
```

OleDbPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class OleDbPermission
   extends System.Data.Common.DBDataPermission
```

Public Constructors

```
public OleDbPermission();
public OleDbPermission(
   System.Security.Permissions.PermissionState state
);
public OleDbPermission(
   System.Security.Permissions.PermissionState state,
   boolean allowBlankPassword
);
```

Public Properties

```
public System.String get_Provider();
public void set_Provider(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement securityElement
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

Protected Methods

```
protected System.Data.Common.DBDataPermission CreateInstance();
```

OleDbPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class OleDbPermissionAttribute
   extends System.Data.Common.DBDataPermissionAttribute
```

Public Constructors

```
public OleDbPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.String get_Provider();
public void set_Provider(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

OleDbRowUpdatedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class OleDbRowUpdatedEventArgs
   extends System.Data.Common.RowUpdatedEventArgs
```

Public Constructors

```
public OleDbRowUpdatedEventArgs(
   System.Data.DataRow dataRow,
   System.Data.IDbCommand command,
   System.Data.StatementType statementType,
   System.Data.Common.DataTableMapping tableMapping
);
```

Public Properties

```
public System.Data.OleDb.OleDbCommand get_Command();
```

OleDbRowUpdatedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void OleDbRowUpdatedEventHandler(
   System.Object sender,
   System.Data.OleDb.OleDbRowUpdatedEventArgs e
);
```

OleDbRowUpdatingEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class OleDbRowUpdatingEventArgs
   extends System.Data.Common.RowUpdatingEventArgs
```

Public Constructors

```
public OleDbRowUpdatingEventArgs(
   System.Data.DataRow dataRow,
   System.Data.IDbCommand command,
   System.Data.StatementType statementType,
   System.Data.Common.DataTableMapping tableMapping
);
```

Public Properties

```
public System.Data.OleDb.OleDbCommand get_Command();
public void set_Command(
   System.Data.OleDb.OleDbCommand value
);
```

OleDbRowUpdatingEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void OleDbRowUpdatingEventHandler(
   System.Object sender,
   System.Data.OleDb.OleDbRowUpdatingEventArgs e
);
```

OleDbSchemaGuid Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class OleDbSchemaGuid
   extends System.Object
```

Public Constructors

```
public OleDbSchemaGuid();
```

Public Fields

```
public static final System.Guid Assertions;
public static final System.Guid Catalogs;
public static final System.Guid Character_Sets;
public static final System.Guid Check_Constraints;
public static final System.Guid Check_Constraints_By_Table;
public static final System.Guid Collations;
public static final System.Guid Column_Domain_Usage;
public static final System.Guid Column_Privileges;
public static final System.Guid Columns;
public static final System.Guid Constraint_Column_Usage;
public static final System.Guid Constraint_Table_Usage;
public static final System.Guid DbInfoLiterals;
public static final System.Guid Foreign_Keys;
public static final System.Guid Indexes;
public static final System.Guid Key_Column_Usage;
public static final System.Guid Primary_Keys;
public static final System.Guid Procedure_Columns;
public static final System.Guid Procedure_Parameters;
public static final System.Guid Procedures;
public static final System.Guid Provider_Types;
public static final System.Guid Referential_Constraints;
public static final System.Guid Schemata;
public static final System.Guid Sql_Languages;
public static final System.Guid Statistics;
public static final System.Guid Table_Constraints;
public static final System.Guid Table_Privileges;
public static final System.Guid Table_Statistics;
public static final System.Guid Tables;
public static final System.Guid Tables_Info;
public static final System.Guid Translations;
public static final System.Guid Trustee;
public static final System.Guid Usage_Privileges;
public static final System.Guid View_Column_Usage;
public static final System.Guid View_Table_Usage;
public static final System.Guid Views;
```

OleDbTransaction Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class OleDbTransaction
   extends System.MarshalByRefObject
   implements System.Data.IDbTransaction, System.IDisposable
```

Public Properties

```
public System.Data.OleDb.OleDbConnection get_Connection();
public final System.Data.IsolationLevel get_IsolationLevel();
```

Public Methods

```
public System.Data.OleDb.OleDbTransaction Begin();
public System.Data.OleDb.OleDbTransaction Begin(
   System.Data.IsolationLevel isolevel
);
public final void Commit();
public final void Rollback();
```

Protected Methods

```
protected void Finalize();
```

OleDbType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum OleDbType
```

Members

BigInt

Binary

Boolean

BSTR

Char

Currency

Date

DBDate

DBTime

DBTimeStamp

Decimal

Double

Empty

Error

Filetime

Guid

IDispatch

Integer

IUnknown

LongVarBinary

LongVarChar

LongVarWChar

Numeric

PropVariant

Single

SmallInt

TinyInt

UnsignedBigInt

UnsignedInt

UnsignedSmallInt

UnsignedTinyInt

VarBinary

VarChar

Variant

VarNumeric

VarWChar

WChar

System.Data.SqlClient Namespace (Visual J# Syntax)

Classes

SqlClientPermission

SqlClientPermissionAttribute

SqlCommand

SqlCommandBuilder

SqlConnection

SqlDataAdapter

SqlDataReader

SQLDebugging

SqlError

SqlErrorCollection

SqlException

SqlInfoMessageEventArgs

SqlParameter

SqlParameterCollection

SqlRowUpdatedEventArgs

SqlRowUpdatingEventArgs

SqlTransaction

Delegates

SqlInfoMessageEventHandler

SqlRowUpdatedEventHandler

SqlRowUpdatingEventHandler

SqlClientPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SqlClientPermission
   extends System.Data.Common.DBDataPermission
```

Public Constructors

```
public SqlClientPermission();
public SqlClientPermission(
   System.Security.Permissions.PermissionState state
);
public SqlClientPermission(
   System.Security.Permissions.PermissionState state,
   boolean allowBlankPassword
);
```

Protected Methods

```
protected System.Data.Common.DBDataPermission CreateInstance();
```

SqlClientPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class SqlClientPermissionAttribute
   extends System.Data.Common.DBDataPermissionAttribute
```

Public Constructors

```
public SqlClientPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

SqlCommand Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "Microsoft.VSDesigner.Data.VS.SqlCommandDesigne
r, Microsoft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemType = System.Drawing.De
sign.ToolboxItem, ToolboxItemTypeName = "System.Drawing.Design.ToolboxItem, System.Drawing, V
ersion=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class SqlCommand
   extends System.ComponentModel.Component
   implements System.Data.IDbCommand, System.ICloneable
```

Public Constructors

```
public SqlCommand();
public SqlCommand(
   System.String cmdText
);
public SqlCommand(
   System.String cmdText,
   System.Data.SqlClient.SqlConnection connection
);
public SqlCommand(
   System.String cmdText,
   System.Data.SqlClient.SqlConnection connection,
   System.Data.SqlClient.SqlTransaction transaction
);
```

Public Properties

```
public final System.String get_CommandText();
public final void set_CommandText(
   System.String value
);
public final int get_CommandTimeout();
public final void set_CommandTimeout(
   int value
);
public final System.Data.CommandType get_CommandType();
public final void set_CommandType(
   System.Data.CommandType value
);
public System.Data.SqlClient.SqlConnection get_Connection();
public void set_Connection(
   System.Data.SqlClient.SqlConnection value
);
public boolean get_DesignTimeVisible();
public void set_DesignTimeVisible(
   boolean value
);
public System.Data.SqlClient.SqlParameterCollection get_Parameters();
public System.Data.SqlClient.SqlTransaction get_Transaction();
public void set_Transaction(
   System.Data.SqlClient.SqlTransaction value
);
public final System.Data.UpdateRowSource get_UpdatedRowSource();
```

```
public final void set_UpdatedRowSource(
   System.Data.UpdateRowSource value
);
```

Public Methods

```
public final void Cancel();
public System.Data.SqlClient.SqlParameter CreateParameter();
public final int ExecuteNonQuery();
public System.Data.SqlClient.SqlDataReader ExecuteReader();
public System.Data.SqlClient.SqlDataReader ExecuteReader(
   System.Data.CommandBehavior behavior
);
public final System.Object ExecuteScalar();
public System.Xml.XmlReader ExecuteXmlReader();
public final void Prepare();
public void ResetCommandTimeout();
```

SqlCommandBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlCommandBuilder
   extends System.ComponentModel.Component
```

Public Constructors

```
public SqlCommandBuilder();
public SqlCommandBuilder(
   System.Data.SqlClient.SqlDataAdapter adapter
);
```

Public Properties

```
public System.Data.SqlClient.SqlDataAdapter get_DataAdapter();
public void set_DataAdapter(
   System.Data.SqlClient.SqlDataAdapter value
);
public System.String get_QuotePrefix();
public void set_QuotePrefix(
   System.String value
);
public System.String get_QuoteSuffix();
public void set_QuoteSuffix(
   System.String value
);
```

Public Methods

```
public static void DeriveParameters(
   System.Data.SqlClient.SqlCommand command
);
public System.Data.SqlClient.SqlCommand GetDeleteCommand();
public System.Data.SqlClient.SqlCommand GetInsertCommand();
public System.Data.SqlClient.SqlCommand GetUpdateCommand();
public void RefreshSchema();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

SqlConnection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "InfoMessage") */
public class SqlConnection
   extends System.ComponentModel.Component
   implements System.Data.IDbConnection, System.ICloneable
```

Public Constructors

```
public SqlConnection();
public SqlConnection(
   System.String connectionString
);
```

Public Properties

```
public final System.String get_ConnectionString();
public final void set_ConnectionString(
   System.String value
);
public final int get_ConnectionTimeout();
public final System.String get_Database();
public System.String get_DataSource();
public int get_PacketSize();
public System.String get_ServerVersion();
public final System.Data.ConnectionState get_State();
public System.String get_WorkstationId();
```

Public Methods

```
public System.Data.SqlClient.SqlTransaction BeginTransaction();
public System.Data.SqlClient.SqlTransaction BeginTransaction(
   System.Data.IsolationLevel iso
);
public System.Data.SqlClient.SqlTransaction BeginTransaction(
   System.String transactionName
);
public System.Data.SqlClient.SqlTransaction BeginTransaction(
   System.Data.IsolationLevel iso,
   System.String transactionName
);
public final void ChangeDatabase(
   System.String database
);
public final void Close();
public System.Data.SqlClient.SqlCommand CreateCommand();
public final void Open();
```

Public Events

```
public void add_InfoMessage(
   System.Data.SqlClient.SqlInfoMessageEventHandler value
);
```

```
public void remove_InfoMessage(
   System.Data.SqlClient.SqlInfoMessageEventHandler value
);
public void add_StateChange(
   System.Data.StateChangeEventHandler value
);
public void remove_StateChange(
   System.Data.StateChangeEventHandler value
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

SqlDataAdapter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "Microsoft.VSDesigner.Data.VS.SqlDataAdapterDes
igner, Microsoft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d5
0a3a") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "Microsoft.VS
Designer.Data.VS.SqlDataAdapterToolboxItem, Microsoft.VSDesigner, Version=7.0.3300.0, Culture
=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "RowUpdated") */
public class SqlDataAdapter
   extends System.Data.Common.DbDataAdapter
   implements System.Data.IDbDataAdapter
```

Public Constructors

```
public SqlDataAdapter();
public SqlDataAdapter(
   System.Data.SqlClient.SqlCommand selectCommand
);
public SqlDataAdapter(
   System.String selectCommandText,
   System.Data.SqlClient.SqlConnection selectConnection
);
public SqlDataAdapter(
   System.String selectCommandText,
   System.String selectConnectionString
);
```

Public Properties

```
public System.Data.SqlClient.SqlCommand get_DeleteCommand();
public void set_DeleteCommand(
   System.Data.SqlClient.SqlCommand value
);
public System.Data.SqlClient.SqlCommand get_InsertCommand();
public void set_InsertCommand(
   System.Data.SqlClient.SqlCommand value
);
public System.Data.SqlClient.SqlCommand get_SelectCommand();
public void set_SelectCommand(
   System.Data.SqlClient.SqlCommand value
);
public System.Data.SqlClient.SqlCommand get_UpdateCommand();
public void set_UpdateCommand(
   System.Data.SqlClient.SqlCommand value
);
```

Public Events

```
public void add_RowUpdated(
   System.Data.SqlClient.SqlRowUpdatedEventHandler value
);
public void remove_RowUpdated(
   System.Data.SqlClient.SqlRowUpdatedEventHandler value
```

```
);
public void add_RowUpdating(
   System.Data.SqlClient.SqlRowUpdatingEventHandler value
);
public void remove_RowUpdating(
   System.Data.SqlClient.SqlRowUpdatingEventHandler value
);
```

Protected Methods

```
protected System.Data.Common.RowUpdatedEventArgs CreateRowUpdatedEvent(
   System.Data.DataRow dataRow,
   System.Data.IDbCommand command,
   System.Data.StatementType statementType,
   System.Data.Common.DataTableMapping tableMapping
);
protected System.Data.Common.RowUpdatingEventArgs CreateRowUpdatingEvent(
   System.Data.DataRow dataRow,
   System.Data.IDbCommand command,
   System.Data.StatementType statementType,
   System.Data.Common.DataTableMapping tableMapping
);
protected void OnRowUpdated(
   System.Data.Common.RowUpdatedEventArgs value
);
protected void OnRowUpdating(
   System.Data.Common.RowUpdatingEventArgs value
);
```

SqlDataReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SqlDataReader
   extends System.MarshalByRefObject
   implements System.Collections.IEnumerable, System.Data.IDataReader, System.IDisposable, Sy
stem.Data.IDataRecord
```

Public Properties

```
public final int get_Depth();
public final int get_FieldCount();
public final boolean get_IsClosed();
public final System.Object get_Item(
   int i
);
public final System.Object get_Item(
   System.String name
);
public final int get_RecordsAffected();
```

Public Methods

```
public final void Close();
public final boolean GetBoolean(
   int i
);
public final ubyte GetByte(
   int i
);
public final long GetBytes(
   int i,
   long dataIndex,
   ubyte[] buffer,
   int bufferIndex,
   int length
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public final char GetChar(
   int i
);
public final long GetChars(
   int i,
   long dataIndex,
   char[] buffer,
   int bufferIndex,
   int length
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public final System.Data.IDataReader GetData(
   int i
);
public final System.String GetDataTypeName(
   int i
);
public final System.DateTime GetDateTime(
   int i
```

```
);
public final System.Decimal GetDecimal(
   int i
);
public final double GetDouble(
   int i
);
public final System.Type GetFieldType(
   int i
);
public final float GetFloat(
   int i
);
public final System.Guid GetGuid(
   int i
);
public final short GetInt16(
   int i
);
public final int GetInt32(
   int i
);
public final long GetInt64(
   int i
);
public final System.String GetName(
   int i
);
public final int GetOrdinal(
   System.String name
);
public final System.Data.DataTable GetSchemaTable();
public System.Data.SqlTypes.SqlBinary GetSqlBinary(
   int i
);
public System.Data.SqlTypes.SqlBoolean GetSqlBoolean(
   int i
);
public System.Data.SqlTypes.SqlByte GetSqlByte(
   int i
);
public System.Data.SqlTypes.SqlDateTime GetSqlDateTime(
   int i
);
public System.Data.SqlTypes.SqlDecimal GetSqlDecimal(
   int i
);
public System.Data.SqlTypes.SqlDouble GetSqlDouble(
   int i
);
public System.Data.SqlTypes.SqlGuid GetSqlGuid(
   int i
);
public System.Data.SqlTypes.SqlInt16 GetSqlInt16(
   int i
);
public System.Data.SqlTypes.SqlInt32 GetSqlInt32(
   int i
);
public System.Data.SqlTypes.SqlInt64 GetSqlInt64(
   int i
);
public System.Data.SqlTypes.SqlMoney GetSqlMoney(
   int i
);
public System.Data.SqlTypes.SqlSingle GetSqlSingle(
   int i
);
```

```
public System.Data.SqlTypes.SqlString GetSqlString(
   int i
);
public System.Object GetSqlValue(
   int i
);
public int GetSqlValues(
   System.Object[] values
);
public final System.String GetString(
   int i
);
public final System.Object GetValue(
   int i
);
public final int GetValues(
   System.Object[] values
);
public final boolean IsDBNull(
   int i
);
public final boolean NextResult();
public final boolean Read();
```

SQLDebugging Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = None) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "afef65ad-4577-447a-a148-
83acadd3d4b9") */
public class SQLDebugging
   extends System.Object
   implements System.Data.SqlClient.ISQLDebug
```

Public Constructors

```
public SQLDebugging();
```

SqlError Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SqlError
   extends System.Object
```

Public Properties

```
public ubyte get_Class();
public int get_LineNumber();
public System.String get_Message();
public int get_Number();
public System.String get_Procedure();
public System.String get_Server();
public System.String get_Source();
public ubyte get_State();
```

Public Methods

```
public System.String ToString();
```

SqlErrorCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.ListBindableAttribute(ListBindable = false) */
public class SqlErrorCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public System.Data.SqlClient.SqlError get_Item(
   int index
);
```

Public Methods

```
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
```

SqlException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SqlException
   extends System.SystemException
```

Public Properties

```
public ubyte get_Class();
public System.Data.SqlClient.SqlErrorCollection get_Errors();
public int get_LineNumber();
public System.String get_Message();
public int get_Number();
public System.String get_Procedure();
public System.String get_Server();
public System.String get_Source();
public ubyte get_State();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo si,
   System.Runtime.Serialization.StreamingContext context
);
```

SqlInfoMessageEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlInfoMessageEventArgs
   extends System.EventArgs
```

Public Properties

```
public System.Data.SqlClient.SqlErrorCollection get_Errors();
public System.String get_Message();
public System.String get_Source();
```

Public Methods

```
public System.String ToString();
```

SqlInfoMessageEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void SqlInfoMessageEventHandler(
   System.Object sender,
   System.Data.SqlClient.SqlInfoMessageEventArgs e
);
```

SqlParameter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Data.
SqlClient.SqlParameterConverter, System.Data, Version=1.0.3300.0, Culture=neutral, PublicKeyT
oken=b77a5c561934e089") */
public class SqlParameter
   extends System.MarshalByRefObject
   implements System.Data.IDbDataParameter, System.Data.IDataParameter, System.ICloneable
```

Public Constructors

```
public SqlParameter();
public SqlParameter(
   System.String parameterName,
   System.Data.SqlDbType dbType
);
public SqlParameter(
   System.String parameterName,
   System.Object value
);
public SqlParameter(
   System.String parameterName,
   System.Data.SqlDbType dbType,
   int size
);
public SqlParameter(
   System.String parameterName,
   System.Data.SqlDbType dbType,
   int size,
   System.String sourceColumn
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public SqlParameter(
   System.String parameterName,
   System.Data.SqlDbType dbType,
   int size,
   System.Data.ParameterDirection direction,
   boolean isNullable,
   ubyte precision,
   ubyte scale,
   System.String sourceColumn,
   System.Data.DataRowVersion sourceVersion,
   System.Object value
);
```

Public Properties

```
public final System.Data.DbType get_DbType();
public final void set_DbType(
   System.Data.DbType value
);
public final System.Data.ParameterDirection get_Direction();
public final void set_Direction(
   System.Data.ParameterDirection value
);
public final boolean get_IsNullable();
public void set_IsNullable(
```

```
   boolean value
);
public int get_Offset();
public void set_Offset(
   int value
);
public final System.String get_ParameterName();
public final void set_ParameterName(
   System.String value
);
public final ubyte get_Precision();
public final void set_Precision(
   ubyte value
);
public final ubyte get_Scale();
public final void set_Scale(
   ubyte value
);
public final int get_Size();
public final void set_Size(
   int value
);
public final System.String get_SourceColumn();
public final void set_SourceColumn(
   System.String value
);
public final System.Data.DataRowVersion get_SourceVersion();
public final void set_SourceVersion(
   System.Data.DataRowVersion value
);
public System.Data.SqlDbType get_SqlDbType();
public void set_SqlDbType(
   System.Data.SqlDbType value
);
public final System.Object get_Value();
public final void set_Value(
   System.Object value
);
```

Public Methods

```
public System.String ToString();
```

SqlParameterCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "Microsoft.VSDesigner.Data.Design.DBParametersEditor, Microsoft.
VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.ListBindableAttribute(ListBindable = false) */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SqlParameterCollection
   extends System.MarshalByRefObject
   implements System.Data.IDataParameterCollection, System.Collections.IList, System.Collecti
ons.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public System.Data.SqlClient.SqlParameter get_Item(
   int index
);
public System.Data.SqlClient.SqlParameter get_Item(
   System.String parameterName
);
public void set_Item(
   int index,
   System.Data.SqlClient.SqlParameter value
);
public void set_Item(
   System.String parameterName,
   System.Data.SqlClient.SqlParameter value
);
```

Public Methods

```
public final int Add(
   System.Object value
);
public System.Data.SqlClient.SqlParameter Add(
   System.Data.SqlClient.SqlParameter value
);
public System.Data.SqlClient.SqlParameter Add(
   System.String parameterName,
   System.Data.SqlDbType sqlDbType
);
public System.Data.SqlClient.SqlParameter Add(
   System.String parameterName,
   System.Object value
);
public System.Data.SqlClient.SqlParameter Add(
   System.String parameterName,
   System.Data.SqlDbType sqlDbType,
   int size
);
public System.Data.SqlClient.SqlParameter Add(
   System.String parameterName,
   System.Data.SqlDbType sqlDbType,
   int size,
   System.String sourceColumn
```

```
);
public final void Clear();
public final boolean Contains(
   System.Object value
);
public final boolean Contains(
   System.String value
);
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public final int IndexOf(
   System.Object value
);
public final int IndexOf(
   System.String parameterName
);
public final void Insert(
   int index,
   System.Object value
);
public final void Remove(
   System.Object value
);
public final void RemoveAt(
   int index
);
public final void RemoveAt(
   System.String parameterName
);
```

SqlRowUpdatedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlRowUpdatedEventArgs
   extends System.Data.Common.RowUpdatedEventArgs
```

Public Constructors

```
public SqlRowUpdatedEventArgs(
   System.Data.DataRow row,
   System.Data.IDbCommand command,
   System.Data.StatementType statementType,
   System.Data.Common.DataTableMapping tableMapping
);
```

Public Properties

```
public System.Data.SqlClient.SqlCommand get_Command();
```

SqlRowUpdatedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void SqlRowUpdatedEventHandler(
   System.Object sender,
   System.Data.SqlClient.SqlRowUpdatedEventArgs e
);
```

SqlRowUpdatingEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlRowUpdatingEventArgs
   extends System.Data.Common.RowUpdatingEventArgs
```

Public Constructors

```
public SqlRowUpdatingEventArgs(
   System.Data.DataRow row,
   System.Data.IDbCommand command,
   System.Data.StatementType statementType,
   System.Data.Common.DataTableMapping tableMapping
);
```

Public Properties

```
public System.Data.SqlClient.SqlCommand get_Command();
public void set_Command(
   System.Data.SqlClient.SqlCommand value
);
```

SqlRowUpdatingEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void SqlRowUpdatingEventHandler(
   System.Object sender,
   System.Data.SqlClient.SqlRowUpdatingEventArgs e
);
```

SqlTransaction Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlTransaction
   extends System.MarshalByRefObject
   implements System.Data.IDbTransaction, System.IDisposable
```

Public Properties

```
public System.Data.SqlClient.SqlConnection get_Connection();
public final System.Data.IsolationLevel get_IsolationLevel();
```

Public Methods

```
public final void Commit();
public final void Dispose();
public final void Rollback();
public void Rollback(
   System.String transactionName
);
public void Save(
   System.String savePointName
);
```

System.Data.SqlTypes Namespace (Visual J# Syntax)

Classes

SqlNullValueException

SqlTruncateException

SqlTypeException

Interfaces

INullable

Structures

SqlBinary

SqlBoolean

SqlByte

SqlDateTime

SqlDecimal

SqlDouble

SqlGuid

SqlInt16

SqlInt32

SqlInt64

SqlMoney

SqlSingle

SqlString

Enumerations

SqlCompareOptions

INullable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface INullable
```

Public Properties

```
boolean get_IsNull();
```

SqlBinary Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SqlBinary
   extends System.ValueType
   implements System.Data.SqlTypes.INullable, System.IComparable
```

Public Constructors

```
public SqlBinary(
   ubyte[] value
);
```

Public Properties

```
public final boolean get_IsNull();
public ubyte get_Item(
   int index
);
public int get_Length();
public ubyte[] get_Value();
```

Public Methods

```
public final int CompareTo(
   System.Object value
);
public static System.Data.SqlTypes.SqlBinary Concat(
   System.Data.SqlTypes.SqlBinary x,
   System.Data.SqlTypes.SqlBinary y
);
public boolean Equals(
   System.Object value
);
public static System.Data.SqlTypes.SqlBoolean Equals(
   System.Data.SqlTypes.SqlBinary x,
   System.Data.SqlTypes.SqlBinary y
);
public int GetHashCode();
public static System.Data.SqlTypes.SqlBoolean GreaterThan(
   System.Data.SqlTypes.SqlBinary x,
   System.Data.SqlTypes.SqlBinary y
);
public static System.Data.SqlTypes.SqlBoolean GreaterThanOrEqual(
   System.Data.SqlTypes.SqlBinary x,
   System.Data.SqlTypes.SqlBinary y
);
public static System.Data.SqlTypes.SqlBoolean LessThan(
   System.Data.SqlTypes.SqlBinary x,
   System.Data.SqlTypes.SqlBinary y
);
public static System.Data.SqlTypes.SqlBoolean LessThanOrEqual(
   System.Data.SqlTypes.SqlBinary x,
   System.Data.SqlTypes.SqlBinary y
);
```

```
public static System.Data.SqlTypes.SqlBoolean NotEquals(
   System.Data.SqlTypes.SqlBinary x,
   System.Data.SqlTypes.SqlBinary y
);
public System.Data.SqlTypes.SqlGuid ToSqlGuid();
public System.String ToString();
```

Public Operators

```
public static System.Data.SqlTypes.SqlBinary op_Addition(
   System.Data.SqlTypes.SqlBinary x,
   System.Data.SqlTypes.SqlBinary y
);
public static System.Data.SqlTypes.SqlBoolean op_Equality(
   System.Data.SqlTypes.SqlBinary x,
   System.Data.SqlTypes.SqlBinary y
);
public static ubyte[] op_Explicit(
   System.Data.SqlTypes.SqlBinary x
);
public static System.Data.SqlTypes.SqlBinary op_Explicit(
   System.Data.SqlTypes.SqlGuid x
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThan(
   System.Data.SqlTypes.SqlBinary x,
   System.Data.SqlTypes.SqlBinary y
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThanOrEqual(
   System.Data.SqlTypes.SqlBinary x,
   System.Data.SqlTypes.SqlBinary y
);
public static System.Data.SqlTypes.SqlBinary op_Implicit(
   ubyte[] x
);
public static System.Data.SqlTypes.SqlBoolean op_Inequality(
   System.Data.SqlTypes.SqlBinary x,
   System.Data.SqlTypes.SqlBinary y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThan(
   System.Data.SqlTypes.SqlBinary x,
   System.Data.SqlTypes.SqlBinary y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThanOrEqual(
   System.Data.SqlTypes.SqlBinary x,
   System.Data.SqlTypes.SqlBinary y
);
```

Public Fields

```
public static final System.Data.SqlTypes.SqlBinary Null;
```

SqlBoolean Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlBoolean
   extends System.ValueType
   implements System.Data.SqlTypes.INullable, System.IComparable
```

Public Constructors

```
public SqlBoolean(
   boolean value
);
public SqlBoolean(
   int value
);
```

Public Properties

```
public ubyte get_ByteValue();
public boolean get_IsFalse();
public final boolean get_IsNull();
public boolean get_IsTrue();
public boolean get_Value();
```

Public Methods

```
public static System.Data.SqlTypes.SqlBoolean And(
   System.Data.SqlTypes.SqlBoolean x,
   System.Data.SqlTypes.SqlBoolean y
);
public final int CompareTo(
   System.Object value
);
public boolean Equals(
   System.Object value
);
public static System.Data.SqlTypes.SqlBoolean Equals(
   System.Data.SqlTypes.SqlBoolean x,
   System.Data.SqlTypes.SqlBoolean y
);
public int GetHashCode();
public static System.Data.SqlTypes.SqlBoolean NotEquals(
   System.Data.SqlTypes.SqlBoolean x,
   System.Data.SqlTypes.SqlBoolean y
);
public static System.Data.SqlTypes.SqlBoolean OnesComplement(
   System.Data.SqlTypes.SqlBoolean x
);
public static System.Data.SqlTypes.SqlBoolean Or(
   System.Data.SqlTypes.SqlBoolean x,
   System.Data.SqlTypes.SqlBoolean y
);
public static System.Data.SqlTypes.SqlBoolean Parse(
   System.String s
);
public System.Data.SqlTypes.SqlByte ToSqlByte();
```

```
public System.Data.SqlTypes.SqlDecimal ToSqlDecimal();
public System.Data.SqlTypes.SqlDouble ToSqlDouble();
public System.Data.SqlTypes.SqlInt16 ToSqlInt16();
public System.Data.SqlTypes.SqlInt32 ToSqlInt32();
public System.Data.SqlTypes.SqlInt64 ToSqlInt64();
public System.Data.SqlTypes.SqlMoney ToSqlMoney();
public System.Data.SqlTypes.SqlSingle ToSqlSingle();
public System.Data.SqlTypes.SqlString ToSqlString();
public System.String ToString();
public static System.Data.SqlTypes.SqlBoolean Xor(
   System.Data.SqlTypes.SqlBoolean x,
   System.Data.SqlTypes.SqlBoolean y
);
```

Public Operators

```
public static System.Data.SqlTypes.SqlBoolean op_BitwiseAnd(
   System.Data.SqlTypes.SqlBoolean x,
   System.Data.SqlTypes.SqlBoolean y
);
public static System.Data.SqlTypes.SqlBoolean op_BitwiseOr(
   System.Data.SqlTypes.SqlBoolean x,
   System.Data.SqlTypes.SqlBoolean y
);
public static System.Data.SqlTypes.SqlBoolean op_Equality(
   System.Data.SqlTypes.SqlBoolean x,
   System.Data.SqlTypes.SqlBoolean y
);
public static System.Data.SqlTypes.SqlBoolean op_ExclusiveOr(
   System.Data.SqlTypes.SqlBoolean x,
   System.Data.SqlTypes.SqlBoolean y
);
public static boolean op_Explicit(
   System.Data.SqlTypes.SqlBoolean x
);
public static System.Data.SqlTypes.SqlBoolean op_Explicit(
   System.Data.SqlTypes.SqlByte x
);
public static System.Data.SqlTypes.SqlBoolean op_Explicit(
   System.Data.SqlTypes.SqlDecimal x
);
public static System.Data.SqlTypes.SqlBoolean op_Explicit(
   System.Data.SqlTypes.SqlDouble x
);
public static System.Data.SqlTypes.SqlBoolean op_Explicit(
   System.Data.SqlTypes.SqlInt16 x
);
public static System.Data.SqlTypes.SqlBoolean op_Explicit(
   System.Data.SqlTypes.SqlInt32 x
);
public static System.Data.SqlTypes.SqlBoolean op_Explicit(
   System.Data.SqlTypes.SqlInt64 x
);
public static System.Data.SqlTypes.SqlBoolean op_Explicit(
   System.Data.SqlTypes.SqlMoney x
);
public static System.Data.SqlTypes.SqlBoolean op_Explicit(
   System.Data.SqlTypes.SqlSingle x
);
public static System.Data.SqlTypes.SqlBoolean op_Explicit(
   System.Data.SqlTypes.SqlString x
);
public static boolean op_False(
   System.Data.SqlTypes.SqlBoolean x
);
public static System.Data.SqlTypes.SqlBoolean op_Implicit(
```

```
   boolean x
);
public static System.Data.SqlTypes.SqlBoolean op_Inequality(
   System.Data.SqlTypes.SqlBoolean x,
   System.Data.SqlTypes.SqlBoolean y
);
public static System.Data.SqlTypes.SqlBoolean op_LogicalNot(
   System.Data.SqlTypes.SqlBoolean x
);
public static System.Data.SqlTypes.SqlBoolean op_OnesComplement(
   System.Data.SqlTypes.SqlBoolean x
);
public static boolean op_True(
   System.Data.SqlTypes.SqlBoolean x
);
```

Public Fields

```
public static final System.Data.SqlTypes.SqlBoolean False;
public static final System.Data.SqlTypes.SqlBoolean Null;
public static final System.Data.SqlTypes.SqlBoolean One;
public static final System.Data.SqlTypes.SqlBoolean True;
public static final System.Data.SqlTypes.SqlBoolean Zero;
```

SqlByte Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlByte
   extends System.ValueType
   implements System.Data.SqlTypes.INullable, System.IComparable
```

Public Constructors

```
public SqlByte(
   ubyte value
);
```

Public Properties

```
public final boolean get_IsNull();
public ubyte get_Value();
```

Public Methods

```
public static System.Data.SqlTypes.SqlByte Add(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlByte BitwiseAnd(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlByte BitwiseOr(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public final int CompareTo(
   System.Object value
);
public static System.Data.SqlTypes.SqlByte Divide(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public boolean Equals(
   System.Object value
);
public static System.Data.SqlTypes.SqlBoolean Equals(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public int GetHashCode();
public static System.Data.SqlTypes.SqlBoolean GreaterThan(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlBoolean GreaterThanOrEqual(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlBoolean LessThan(
```

```
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlBoolean LessThanOrEqual(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlByte Mod(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlByte Multiply(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlBoolean NotEquals(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlByte OnesComplement(
   System.Data.SqlTypes.SqlByte x
);
public static System.Data.SqlTypes.SqlByte Parse(
   System.String s
);
public static System.Data.SqlTypes.SqlByte Subtract(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public System.Data.SqlTypes.SqlBoolean ToSqlBoolean();
public System.Data.SqlTypes.SqlDecimal ToSqlDecimal();
public System.Data.SqlTypes.SqlDouble ToSqlDouble();
public System.Data.SqlTypes.SqlInt16 ToSqlInt16();
public System.Data.SqlTypes.SqlInt32 ToSqlInt32();
public System.Data.SqlTypes.SqlInt64 ToSqlInt64();
public System.Data.SqlTypes.SqlMoney ToSqlMoney();
public System.Data.SqlTypes.SqlSingle ToSqlSingle();
public System.Data.SqlTypes.SqlString ToSqlString();
public System.String ToString();
public static System.Data.SqlTypes.SqlByte Xor(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
```

Public Operators

```
public static System.Data.SqlTypes.SqlByte op_Addition(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlByte op_BitwiseAnd(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlByte op_BitwiseOr(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlByte op_Division(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlBoolean op_Equality(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
```

```
public static System.Data.SqlTypes.SqlByte op_ExclusiveOr(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static ubyte op_Explicit(
   System.Data.SqlTypes.SqlByte x
);
public static System.Data.SqlTypes.SqlByte op_Explicit(
   System.Data.SqlTypes.SqlBoolean x
);
public static System.Data.SqlTypes.SqlByte op_Explicit(
   System.Data.SqlTypes.SqlDecimal x
);
public static System.Data.SqlTypes.SqlByte op_Explicit(
   System.Data.SqlTypes.SqlDouble x
);
public static System.Data.SqlTypes.SqlByte op_Explicit(
   System.Data.SqlTypes.SqlInt16 x
);
public static System.Data.SqlTypes.SqlByte op_Explicit(
   System.Data.SqlTypes.SqlInt32 x
);
public static System.Data.SqlTypes.SqlByte op_Explicit(
   System.Data.SqlTypes.SqlInt64 x
);
public static System.Data.SqlTypes.SqlByte op_Explicit(
   System.Data.SqlTypes.SqlMoney x
);
public static System.Data.SqlTypes.SqlByte op_Explicit(
   System.Data.SqlTypes.SqlSingle x
);
public static System.Data.SqlTypes.SqlByte op_Explicit(
   System.Data.SqlTypes.SqlString x
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThan(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThanOrEqual(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlByte op_Implicit(
   ubyte x
);
public static System.Data.SqlTypes.SqlBoolean op_Inequality(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThan(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThanOrEqual(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlByte op_Modulus(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlByte op_Multiply(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
public static System.Data.SqlTypes.SqlByte op_OnesComplement(
   System.Data.SqlTypes.SqlByte x
);
```

```
public static System.Data.SqlTypes.SqlByte op_Subtraction(
   System.Data.SqlTypes.SqlByte x,
   System.Data.SqlTypes.SqlByte y
);
```

Public Fields

```
public static final System.Data.SqlTypes.SqlByte MaxValue;
public static final System.Data.SqlTypes.SqlByte MinValue;
public static final System.Data.SqlTypes.SqlByte Null;
public static final System.Data.SqlTypes.SqlByte Zero;
```

SqlCompareOptions Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum SqlCompareOptions
```

Members

BinarySort

IgnoreCase

IgnoreKanaType

IgnoreNonSpace

IgnoreWidth

None

SqlDateTime Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlDateTime
   extends System.ValueType
   implements System.Data.SqlTypes.INullable, System.IComparable
```

Public Constructors

```
public SqlDateTime(
   System.DateTime value
);
public SqlDateTime(
   int dayTicks,
   int timeTicks
);
public SqlDateTime(
   int year,
   int month,
   int day
);
public SqlDateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second
);
public SqlDateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second,
   double millisecond
);
public SqlDateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second,
   int bilisecond
);
```

Public Properties

```
public int get_DayTicks();
public final boolean get_IsNull();
public int get_TimeTicks();
public System.DateTime get_Value();
```

Public Methods

```
public final int CompareTo(
   System.Object value
);
public boolean Equals(
   System.Object value
);
public static System.Data.SqlTypes.SqlBoolean Equals(
   System.Data.SqlTypes.SqlDateTime x,
   System.Data.SqlTypes.SqlDateTime y
);
public int GetHashCode();
public static System.Data.SqlTypes.SqlBoolean GreaterThan(
   System.Data.SqlTypes.SqlDateTime x,
   System.Data.SqlTypes.SqlDateTime y
);
public static System.Data.SqlTypes.SqlBoolean GreaterThanOrEqual(
   System.Data.SqlTypes.SqlDateTime x,
   System.Data.SqlTypes.SqlDateTime y
);
public static System.Data.SqlTypes.SqlBoolean LessThan(
   System.Data.SqlTypes.SqlDateTime x,
   System.Data.SqlTypes.SqlDateTime y
);
public static System.Data.SqlTypes.SqlBoolean LessThanOrEqual(
   System.Data.SqlTypes.SqlDateTime x,
   System.Data.SqlTypes.SqlDateTime y
);
public static System.Data.SqlTypes.SqlBoolean NotEquals(
   System.Data.SqlTypes.SqlDateTime x,
   System.Data.SqlTypes.SqlDateTime y
);
public static System.Data.SqlTypes.SqlDateTime Parse(
   System.String s
);
public System.Data.SqlTypes.SqlString ToSqlString();
public System.String ToString();
```

Public Operators

```
public static System.Data.SqlTypes.SqlDateTime op_Addition(
   System.Data.SqlTypes.SqlDateTime x,
   System.TimeSpan t
);
public static System.Data.SqlTypes.SqlBoolean op_Equality(
   System.Data.SqlTypes.SqlDateTime x,
   System.Data.SqlTypes.SqlDateTime y
);
public static System.Data.SqlTypes.SqlDateTime op_Explicit(
   System.Data.SqlTypes.SqlString x
);
public static System.DateTime op_Explicit(
   System.Data.SqlTypes.SqlDateTime x
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThan(
   System.Data.SqlTypes.SqlDateTime x,
   System.Data.SqlTypes.SqlDateTime y
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThanOrEqual(
   System.Data.SqlTypes.SqlDateTime x,
   System.Data.SqlTypes.SqlDateTime y
);
public static System.Data.SqlTypes.SqlDateTime op_Implicit(
   System.DateTime value
);
public static System.Data.SqlTypes.SqlBoolean op_Inequality(
   System.Data.SqlTypes.SqlDateTime x,
```

```
   System.Data.SqlTypes.SqlDateTime y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThan(
   System.Data.SqlTypes.SqlDateTime x,
   System.Data.SqlTypes.SqlDateTime y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThanOrEqual(
   System.Data.SqlTypes.SqlDateTime x,
   System.Data.SqlTypes.SqlDateTime y
);
public static System.Data.SqlTypes.SqlDateTime op_Subtraction(
   System.Data.SqlTypes.SqlDateTime x,
   System.TimeSpan t
);
```

Public Fields

```
public static final System.Data.SqlTypes.SqlDateTime MaxValue;
public static final System.Data.SqlTypes.SqlDateTime MinValue;
public static final System.Data.SqlTypes.SqlDateTime Null;
public static final int SQLTicksPerHour;
public static final int SQLTicksPerMinute;
public static final int SQLTicksPerSecond;
```

SqlDecimal Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlDecimal
   extends System.ValueType
   implements System.Data.SqlTypes.INullable, System.IComparable
```

Public Constructors

```
public SqlDecimal(
   double dVal
);
public SqlDecimal(
   int value
);
public SqlDecimal(
   long value
);
public SqlDecimal(
   System.Decimal value
);
public SqlDecimal(
   ubyte bPrecision,
   ubyte bScale,
   boolean fPositive,
   int[] bits
);
public SqlDecimal(
   ubyte bPrecision,
   ubyte bScale,
   boolean fPositive,
   int data1,
   int data2,
   int data3,
   int data4
);
```

Public Properties

```
public ubyte[] get_BinData();
public int[] get_Data();
public final boolean get_IsNull();
public boolean get_IsPositive();
public ubyte get_Precision();
public ubyte get_Scale();
public System.Decimal get_Value();
```

Public Methods

```
public static System.Data.SqlTypes.SqlDecimal Abs(
   System.Data.SqlTypes.SqlDecimal n
);
public static System.Data.SqlTypes.SqlDecimal Add(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
```

```
public static System.Data.SqlTypes.SqlDecimal AdjustScale(
   System.Data.SqlTypes.SqlDecimal n,
   int digits,
   boolean fRound
);
public static System.Data.SqlTypes.SqlDecimal Ceiling(
   System.Data.SqlTypes.SqlDecimal n
);
public final int CompareTo(
   System.Object value
);
public static System.Data.SqlTypes.SqlDecimal ConvertToPrecScale(
   System.Data.SqlTypes.SqlDecimal n,
   int precision,
   int scale
);
public static System.Data.SqlTypes.SqlDecimal Divide(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public boolean Equals(
   System.Object value
);
public static System.Data.SqlTypes.SqlBoolean Equals(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlDecimal Floor(
   System.Data.SqlTypes.SqlDecimal n
);
public int GetHashCode();
public static System.Data.SqlTypes.SqlBoolean GreaterThan(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlBoolean GreaterThanOrEqual(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlBoolean LessThan(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlBoolean LessThanOrEqual(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlDecimal Multiply(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlBoolean NotEquals(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlDecimal Parse(
   System.String s
);
public static System.Data.SqlTypes.SqlDecimal Power(
   System.Data.SqlTypes.SqlDecimal n,
   double exp
);
public static System.Data.SqlTypes.SqlDecimal Round(
   System.Data.SqlTypes.SqlDecimal n,
   int position
);
public static System.Data.SqlTypes.SqlInt32 Sign(
   System.Data.SqlTypes.SqlDecimal n
```

```
);
public static System.Data.SqlTypes.SqlDecimal Subtract(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public double ToDouble();
public System.Data.SqlTypes.SqlBoolean ToSqlBoolean();
public System.Data.SqlTypes.SqlByte ToSqlByte();
public System.Data.SqlTypes.SqlDouble ToSqlDouble();
public System.Data.SqlTypes.SqlInt16 ToSqlInt16();
public System.Data.SqlTypes.SqlInt32 ToSqlInt32();
public System.Data.SqlTypes.SqlInt64 ToSqlInt64();
public System.Data.SqlTypes.SqlMoney ToSqlMoney();
public System.Data.SqlTypes.SqlSingle ToSqlSingle();
public System.Data.SqlTypes.SqlString ToSqlString();
public System.String ToString();
public static System.Data.SqlTypes.SqlDecimal Truncate(
   System.Data.SqlTypes.SqlDecimal n,
   int position
);
```

Public Operators

```
public static System.Data.SqlTypes.SqlDecimal op_Addition(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlDecimal op_Division(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlBoolean op_Equality(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlDecimal op_Explicit(
   System.Data.SqlTypes.SqlBoolean x
);
public static System.Data.SqlTypes.SqlDecimal op_Explicit(
   System.Data.SqlTypes.SqlDouble x
);
public static System.Data.SqlTypes.SqlDecimal op_Explicit(
   System.Data.SqlTypes.SqlSingle x
);
public static System.Data.SqlTypes.SqlDecimal op_Explicit(
   System.Data.SqlTypes.SqlString x
);
public static System.Decimal op_Explicit(
   System.Data.SqlTypes.SqlDecimal x
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThan(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThanOrEqual(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlDecimal op_Implicit(
   System.Data.SqlTypes.SqlByte x
);
public static System.Data.SqlTypes.SqlDecimal op_Implicit(
   System.Data.SqlTypes.SqlInt16 x
);
public static System.Data.SqlTypes.SqlDecimal op_Implicit(
   System.Data.SqlTypes.SqlInt32 x
```

```
);
public static System.Data.SqlTypes.SqlDecimal op_Implicit(
   System.Data.SqlTypes.SqlInt64 x
);
public static System.Data.SqlTypes.SqlDecimal op_Implicit(
   System.Data.SqlTypes.SqlMoney x
);
public static System.Data.SqlTypes.SqlDecimal op_Implicit(
   System.Decimal x
);
public static System.Data.SqlTypes.SqlBoolean op_Inequality(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThan(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThanOrEqual(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlDecimal op_Multiply(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlDecimal op_Subtraction(
   System.Data.SqlTypes.SqlDecimal x,
   System.Data.SqlTypes.SqlDecimal y
);
public static System.Data.SqlTypes.SqlDecimal op_UnaryNegation(
   System.Data.SqlTypes.SqlDecimal x
);
```

Public Fields

```
public static final ubyte MaxPrecision;
public static final ubyte MaxScale;
public static final System.Data.SqlTypes.SqlDecimal MaxValue;
public static final System.Data.SqlTypes.SqlDecimal MinValue;
public static final System.Data.SqlTypes.SqlDecimal Null;
```

SqlDouble Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlDouble
   extends System.ValueType
   implements System.Data.SqlTypes.INullable, System.IComparable
```

Public Constructors

```
public SqlDouble(
   double value
);
```

Public Properties

```
public final boolean get_IsNull();
public double get_Value();
```

Public Methods

```
public static System.Data.SqlTypes.SqlDouble Add(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public final int CompareTo(
   System.Object value
);
public static System.Data.SqlTypes.SqlDouble Divide(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public boolean Equals(
   System.Object value
);
public static System.Data.SqlTypes.SqlBoolean Equals(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public int GetHashCode();
public static System.Data.SqlTypes.SqlBoolean GreaterThan(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlBoolean GreaterThanOrEqual(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlBoolean LessThan(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlBoolean LessThanOrEqual(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlDouble Multiply(
```

```
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlBoolean NotEquals(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlDouble Parse(
   System.String s
);
public static System.Data.SqlTypes.SqlDouble Subtract(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public System.Data.SqlTypes.SqlBoolean ToSqlBoolean();
public System.Data.SqlTypes.SqlByte ToSqlByte();
public System.Data.SqlTypes.SqlDecimal ToSqlDecimal();
public System.Data.SqlTypes.SqlInt16 ToSqlInt16();
public System.Data.SqlTypes.SqlInt32 ToSqlInt32();
public System.Data.SqlTypes.SqlInt64 ToSqlInt64();
public System.Data.SqlTypes.SqlMoney ToSqlMoney();
public System.Data.SqlTypes.SqlSingle ToSqlSingle();
public System.Data.SqlTypes.SqlString ToSqlString();
public System.String ToString();
```

Public Operators

```
public static System.Data.SqlTypes.SqlDouble op_Addition(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlDouble op_Division(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlBoolean op_Equality(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static double op_Explicit(
   System.Data.SqlTypes.SqlDouble x
);
public static System.Data.SqlTypes.SqlDouble op_Explicit(
   System.Data.SqlTypes.SqlBoolean x
);
public static System.Data.SqlTypes.SqlDouble op_Explicit(
   System.Data.SqlTypes.SqlString x
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThan(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThanOrEqual(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlDouble op_Implicit(
   double x
);
public static System.Data.SqlTypes.SqlDouble op_Implicit(
   System.Data.SqlTypes.SqlByte x
);
public static System.Data.SqlTypes.SqlDouble op_Implicit(
   System.Data.SqlTypes.SqlDecimal x
);
public static System.Data.SqlTypes.SqlDouble op_Implicit(
```

```
   System.Data.SqlTypes.SqlInt16 x
);
public static System.Data.SqlTypes.SqlDouble op_Implicit(
   System.Data.SqlTypes.SqlInt32 x
);
public static System.Data.SqlTypes.SqlDouble op_Implicit(
   System.Data.SqlTypes.SqlInt64 x
);
public static System.Data.SqlTypes.SqlDouble op_Implicit(
   System.Data.SqlTypes.SqlMoney x
);
public static System.Data.SqlTypes.SqlDouble op_Implicit(
   System.Data.SqlTypes.SqlSingle x
);
public static System.Data.SqlTypes.SqlBoolean op_Inequality(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThan(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThanOrEqual(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlDouble op_Multiply(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlDouble op_Subtraction(
   System.Data.SqlTypes.SqlDouble x,
   System.Data.SqlTypes.SqlDouble y
);
public static System.Data.SqlTypes.SqlDouble op_UnaryNegation(
   System.Data.SqlTypes.SqlDouble x
);
```

Public Fields

```
public static final System.Data.SqlTypes.SqlDouble MaxValue;
public static final System.Data.SqlTypes.SqlDouble MinValue;
public static final System.Data.SqlTypes.SqlDouble Null;
public static final System.Data.SqlTypes.SqlDouble Zero;
```

SqlGuid Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlGuid
   extends System.ValueType
   implements System.Data.SqlTypes.INullable, System.IComparable
```

Public Constructors

```
public SqlGuid(
   ubyte[] value
);
public SqlGuid(
   System.Guid g
);
public SqlGuid(
   System.String s
);
public SqlGuid(
   int a,
   short b,
   short c,
   ubyte d,
   ubyte e,
   ubyte f,
   ubyte g,
   ubyte h,
   ubyte i,
   ubyte j,
   ubyte k
);
```

Public Properties

```
public final boolean get_IsNull();
public System.Guid get_Value();
```

Public Methods

```
public final int CompareTo(
   System.Object value
);
public boolean Equals(
   System.Object value
);
public static System.Data.SqlTypes.SqlBoolean Equals(
   System.Data.SqlTypes.SqlGuid x,
   System.Data.SqlTypes.SqlGuid y
);
public int GetHashCode();
public static System.Data.SqlTypes.SqlBoolean GreaterThan(
   System.Data.SqlTypes.SqlGuid x,
   System.Data.SqlTypes.SqlGuid y
);
public static System.Data.SqlTypes.SqlBoolean GreaterThanOrEqual(
   System.Data.SqlTypes.SqlGuid x,
```

```
   System.Data.SqlTypes.SqlGuid y
);
public static System.Data.SqlTypes.SqlBoolean LessThan(
   System.Data.SqlTypes.SqlGuid x,
   System.Data.SqlTypes.SqlGuid y
);
public static System.Data.SqlTypes.SqlBoolean LessThanOrEqual(
   System.Data.SqlTypes.SqlGuid x,
   System.Data.SqlTypes.SqlGuid y
);
public static System.Data.SqlTypes.SqlBoolean NotEquals(
   System.Data.SqlTypes.SqlGuid x,
   System.Data.SqlTypes.SqlGuid y
);
public static System.Data.SqlTypes.SqlGuid Parse(
   System.String s
);
public ubyte[] ToByteArray();
public System.Data.SqlTypes.SqlBinary ToSqlBinary();
public System.Data.SqlTypes.SqlString ToSqlString();
public System.String ToString();
```

Public Operators

```
public static System.Data.SqlTypes.SqlBoolean op_Equality(
   System.Data.SqlTypes.SqlGuid x,
   System.Data.SqlTypes.SqlGuid y
);
public static System.Data.SqlTypes.SqlGuid op_Explicit(
   System.Data.SqlTypes.SqlBinary x
);
public static System.Data.SqlTypes.SqlGuid op_Explicit(
   System.Data.SqlTypes.SqlString x
);
public static System.Guid op_Explicit(
   System.Data.SqlTypes.SqlGuid x
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThan(
   System.Data.SqlTypes.SqlGuid x,
   System.Data.SqlTypes.SqlGuid y
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThanOrEqual(
   System.Data.SqlTypes.SqlGuid x,
   System.Data.SqlTypes.SqlGuid y
);
public static System.Data.SqlTypes.SqlGuid op_Implicit(
   System.Guid x
);
public static System.Data.SqlTypes.SqlBoolean op_Inequality(
   System.Data.SqlTypes.SqlGuid x,
   System.Data.SqlTypes.SqlGuid y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThan(
   System.Data.SqlTypes.SqlGuid x,
   System.Data.SqlTypes.SqlGuid y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThanOrEqual(
   System.Data.SqlTypes.SqlGuid x,
   System.Data.SqlTypes.SqlGuid y
);
```

Public Fields

```
public static final System.Data.SqlTypes.SqlGuid Null;
```

SqlInt16 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlInt16
   extends System.ValueType
   implements System.Data.SqlTypes.INullable, System.IComparable
```

Public Constructors

```
public SqlInt16(
   short value
);
```

Public Properties

```
public final boolean get_IsNull();
public short get_Value();
```

Public Methods

```
public static System.Data.SqlTypes.SqlInt16 Add(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlInt16 BitwiseAnd(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlInt16 BitwiseOr(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public final int CompareTo(
   System.Object value
);
public static System.Data.SqlTypes.SqlInt16 Divide(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public boolean Equals(
   System.Object value
);
public static System.Data.SqlTypes.SqlBoolean Equals(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public int GetHashCode();
public static System.Data.SqlTypes.SqlBoolean GreaterThan(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlBoolean GreaterThanOrEqual(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlBoolean LessThan(
```

```
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlBoolean LessThanOrEqual(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlInt16 Mod(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlInt16 Multiply(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlBoolean NotEquals(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlInt16 OnesComplement(
   System.Data.SqlTypes.SqlInt16 x
);
public static System.Data.SqlTypes.SqlInt16 Parse(
   System.String s
);
public static System.Data.SqlTypes.SqlInt16 Subtract(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public System.Data.SqlTypes.SqlBoolean ToSqlBoolean();
public System.Data.SqlTypes.SqlByte ToSqlByte();
public System.Data.SqlTypes.SqlDecimal ToSqlDecimal();
public System.Data.SqlTypes.SqlDouble ToSqlDouble();
public System.Data.SqlTypes.SqlInt32 ToSqlInt32();
public System.Data.SqlTypes.SqlInt64 ToSqlInt64();
public System.Data.SqlTypes.SqlMoney ToSqlMoney();
public System.Data.SqlTypes.SqlSingle ToSqlSingle();
public System.Data.SqlTypes.SqlString ToSqlString();
public System.String ToString();
public static System.Data.SqlTypes.SqlInt16 Xor(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
```

Public Operators

```
public static System.Data.SqlTypes.SqlInt16 op_Addition(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlInt16 op_BitwiseAnd(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlInt16 op_BitwiseOr(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlInt16 op_Division(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlBoolean op_Equality(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
```

```
public static System.Data.SqlTypes.SqlInt16 op_ExclusiveOr(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static short op_Explicit(
   System.Data.SqlTypes.SqlInt16 x
);
public static System.Data.SqlTypes.SqlInt16 op_Explicit(
   System.Data.SqlTypes.SqlBoolean x
);
public static System.Data.SqlTypes.SqlInt16 op_Explicit(
   System.Data.SqlTypes.SqlDecimal x
);
public static System.Data.SqlTypes.SqlInt16 op_Explicit(
   System.Data.SqlTypes.SqlDouble x
);
public static System.Data.SqlTypes.SqlInt16 op_Explicit(
   System.Data.SqlTypes.SqlInt32 x
);
public static System.Data.SqlTypes.SqlInt16 op_Explicit(
   System.Data.SqlTypes.SqlInt64 x
);
public static System.Data.SqlTypes.SqlInt16 op_Explicit(
   System.Data.SqlTypes.SqlMoney x
);
public static System.Data.SqlTypes.SqlInt16 op_Explicit(
   System.Data.SqlTypes.SqlSingle x
);
public static System.Data.SqlTypes.SqlInt16 op_Explicit(
   System.Data.SqlTypes.SqlString x
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThan(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThanOrEqual(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlInt16 op_Implicit(
   short x
);
public static System.Data.SqlTypes.SqlInt16 op_Implicit(
   System.Data.SqlTypes.SqlByte x
);
public static System.Data.SqlTypes.SqlBoolean op_Inequality(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThan(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThanOrEqual(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlInt16 op_Modulus(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlInt16 op_Multiply(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlInt16 op_OnesComplement(
   System.Data.SqlTypes.SqlInt16 x
);
```

```
public static System.Data.SqlTypes.SqlInt16 op_Subtraction(
   System.Data.SqlTypes.SqlInt16 x,
   System.Data.SqlTypes.SqlInt16 y
);
public static System.Data.SqlTypes.SqlInt16 op_UnaryNegation(
   System.Data.SqlTypes.SqlInt16 x
);
```

Public Fields

```
public static final System.Data.SqlTypes.SqlInt16 MaxValue;
public static final System.Data.SqlTypes.SqlInt16 MinValue;
public static final System.Data.SqlTypes.SqlInt16 Null;
public static final System.Data.SqlTypes.SqlInt16 Zero;
```

SqlInt32 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlInt32
   extends System.ValueType
   implements System.Data.SqlTypes.INullable, System.IComparable
```

Public Constructors

```
public SqlInt32(
   int value
);
```

Public Properties

```
public final boolean get_IsNull();
public int get_Value();
```

Public Methods

```
public static System.Data.SqlTypes.SqlInt32 Add(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlInt32 BitwiseAnd(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlInt32 BitwiseOr(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public final int CompareTo(
   System.Object value
);
public static System.Data.SqlTypes.SqlInt32 Divide(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public boolean Equals(
   System.Object value
);
public static System.Data.SqlTypes.SqlBoolean Equals(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public int GetHashCode();
public static System.Data.SqlTypes.SqlBoolean GreaterThan(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlBoolean GreaterThanOrEqual(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlBoolean LessThan(
```

```
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlBoolean LessThanOrEqual(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlInt32 Mod(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlInt32 Multiply(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlBoolean NotEquals(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlInt32 OnesComplement(
   System.Data.SqlTypes.SqlInt32 x
);
public static System.Data.SqlTypes.SqlInt32 Parse(
   System.String s
);
public static System.Data.SqlTypes.SqlInt32 Subtract(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public System.Data.SqlTypes.SqlBoolean ToSqlBoolean();
public System.Data.SqlTypes.SqlByte ToSqlByte();
public System.Data.SqlTypes.SqlDecimal ToSqlDecimal();
public System.Data.SqlTypes.SqlDouble ToSqlDouble();
public System.Data.SqlTypes.SqlInt16 ToSqlInt16();
public System.Data.SqlTypes.SqlInt64 ToSqlInt64();
public System.Data.SqlTypes.SqlMoney ToSqlMoney();
public System.Data.SqlTypes.SqlSingle ToSqlSingle();
public System.Data.SqlTypes.SqlString ToSqlString();
public System.String ToString();
public static System.Data.SqlTypes.SqlInt32 Xor(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
```

Public Operators

```
public static System.Data.SqlTypes.SqlInt32 op_Addition(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlInt32 op_BitwiseAnd(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlInt32 op_BitwiseOr(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlInt32 op_Division(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlBoolean op_Equality(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
```

```
public static System.Data.SqlTypes.SqlInt32 op_ExclusiveOr(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static int op_Explicit(
   System.Data.SqlTypes.SqlInt32 x
);
public static System.Data.SqlTypes.SqlInt32 op_Explicit(
   System.Data.SqlTypes.SqlBoolean x
);
public static System.Data.SqlTypes.SqlInt32 op_Explicit(
   System.Data.SqlTypes.SqlDecimal x
);
public static System.Data.SqlTypes.SqlInt32 op_Explicit(
   System.Data.SqlTypes.SqlDouble x
);
public static System.Data.SqlTypes.SqlInt32 op_Explicit(
   System.Data.SqlTypes.SqlInt64 x
);
public static System.Data.SqlTypes.SqlInt32 op_Explicit(
   System.Data.SqlTypes.SqlMoney x
);
public static System.Data.SqlTypes.SqlInt32 op_Explicit(
   System.Data.SqlTypes.SqlSingle x
);
public static System.Data.SqlTypes.SqlInt32 op_Explicit(
   System.Data.SqlTypes.SqlString x
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThan(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThanOrEqual(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlInt32 op_Implicit(
   int x
);
public static System.Data.SqlTypes.SqlInt32 op_Implicit(
   System.Data.SqlTypes.SqlByte x
);
public static System.Data.SqlTypes.SqlInt32 op_Implicit(
   System.Data.SqlTypes.SqlInt16 x
);
public static System.Data.SqlTypes.SqlBoolean op_Inequality(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThan(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThanOrEqual(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlInt32 op_Modulus(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlInt32 op_Multiply(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlInt32 op_OnesComplement(
   System.Data.SqlTypes.SqlInt32 x
);
```

```
public static System.Data.SqlTypes.SqlInt32 op_Subtraction(
   System.Data.SqlTypes.SqlInt32 x,
   System.Data.SqlTypes.SqlInt32 y
);
public static System.Data.SqlTypes.SqlInt32 op_UnaryNegation(
   System.Data.SqlTypes.SqlInt32 x
);
```

Public Fields

```
public static final System.Data.SqlTypes.SqlInt32 MaxValue;
public static final System.Data.SqlTypes.SqlInt32 MinValue;
public static final System.Data.SqlTypes.SqlInt32 Null;
public static final System.Data.SqlTypes.SqlInt32 Zero;
```

SqlInt64 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlInt64
   extends System.ValueType
   implements System.Data.SqlTypes.INullable, System.IComparable
```

Public Constructors

```
public SqlInt64(
   long value
);
```

Public Properties

```
public final boolean get_IsNull();
public long get_Value();
```

Public Methods

```
public static System.Data.SqlTypes.SqlInt64 Add(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlInt64 BitwiseAnd(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlInt64 BitwiseOr(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public final int CompareTo(
   System.Object value
);
public static System.Data.SqlTypes.SqlInt64 Divide(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public boolean Equals(
   System.Object value
);
public static System.Data.SqlTypes.SqlBoolean Equals(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public int GetHashCode();
public static System.Data.SqlTypes.SqlBoolean GreaterThan(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlBoolean GreaterThanOrEqual(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlBoolean LessThan(
```

```
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlBoolean LessThanOrEqual(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlInt64 Mod(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlInt64 Multiply(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlBoolean NotEquals(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlInt64 OnesComplement(
   System.Data.SqlTypes.SqlInt64 x
);
public static System.Data.SqlTypes.SqlInt64 Parse(
   System.String s
);
public static System.Data.SqlTypes.SqlInt64 Subtract(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public System.Data.SqlTypes.SqlBoolean ToSqlBoolean();
public System.Data.SqlTypes.SqlByte ToSqlByte();
public System.Data.SqlTypes.SqlDecimal ToSqlDecimal();
public System.Data.SqlTypes.SqlDouble ToSqlDouble();
public System.Data.SqlTypes.SqlInt16 ToSqlInt16();
public System.Data.SqlTypes.SqlInt32 ToSqlInt32();
public System.Data.SqlTypes.SqlMoney ToSqlMoney();
public System.Data.SqlTypes.SqlSingle ToSqlSingle();
public System.Data.SqlTypes.SqlString ToSqlString();
public System.String ToString();
public static System.Data.SqlTypes.SqlInt64 Xor(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
```

Public Operators

```
public static System.Data.SqlTypes.SqlInt64 op_Addition(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlInt64 op_BitwiseAnd(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlInt64 op_BitwiseOr(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlInt64 op_Division(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlBoolean op_Equality(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
```

```
public static System.Data.SqlTypes.SqlInt64 op_ExclusiveOr(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static long op_Explicit(
   System.Data.SqlTypes.SqlInt64 x
);
public static System.Data.SqlTypes.SqlInt64 op_Explicit(
   System.Data.SqlTypes.SqlBoolean x
);
public static System.Data.SqlTypes.SqlInt64 op_Explicit(
   System.Data.SqlTypes.SqlDecimal x
);
public static System.Data.SqlTypes.SqlInt64 op_Explicit(
   System.Data.SqlTypes.SqlDouble x
);
public static System.Data.SqlTypes.SqlInt64 op_Explicit(
   System.Data.SqlTypes.SqlMoney x
);
public static System.Data.SqlTypes.SqlInt64 op_Explicit(
   System.Data.SqlTypes.SqlSingle x
);
public static System.Data.SqlTypes.SqlInt64 op_Explicit(
   System.Data.SqlTypes.SqlString x
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThan(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThanOrEqual(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlInt64 op_Implicit(
   long x
);
public static System.Data.SqlTypes.SqlInt64 op_Implicit(
   System.Data.SqlTypes.SqlByte x
);
public static System.Data.SqlTypes.SqlInt64 op_Implicit(
   System.Data.SqlTypes.SqlInt16 x
);
public static System.Data.SqlTypes.SqlInt64 op_Implicit(
   System.Data.SqlTypes.SqlInt32 x
);
public static System.Data.SqlTypes.SqlBoolean op_Inequality(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThan(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThanOrEqual(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlInt64 op_Modulus(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlInt64 op_Multiply(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlInt64 op_OnesComplement(
   System.Data.SqlTypes.SqlInt64 x
);
```

```
public static System.Data.SqlTypes.SqlInt64 op_Subtraction(
   System.Data.SqlTypes.SqlInt64 x,
   System.Data.SqlTypes.SqlInt64 y
);
public static System.Data.SqlTypes.SqlInt64 op_UnaryNegation(
   System.Data.SqlTypes.SqlInt64 x
);
```

Public Fields

```
public static final System.Data.SqlTypes.SqlInt64 MaxValue;
public static final System.Data.SqlTypes.SqlInt64 MinValue;
public static final System.Data.SqlTypes.SqlInt64 Null;
public static final System.Data.SqlTypes.SqlInt64 Zero;
```

SqlMoney Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlMoney
   extends System.ValueType
   implements System.Data.SqlTypes.INullable, System.IComparable
```

Public Constructors

```
public SqlMoney(
   double value
);
public SqlMoney(
   int value
);
public SqlMoney(
   long value
);
public SqlMoney(
   System.Decimal value
);
```

Public Properties

```
public final boolean get_IsNull();
public System.Decimal get_Value();
```

Public Methods

```
public static System.Data.SqlTypes.SqlMoney Add(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public final int CompareTo(
   System.Object value
);
public static System.Data.SqlTypes.SqlMoney Divide(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public boolean Equals(
   System.Object value
);
public static System.Data.SqlTypes.SqlBoolean Equals(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public int GetHashCode();
public static System.Data.SqlTypes.SqlBoolean GreaterThan(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlBoolean GreaterThanOrEqual(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
```

```
public static System.Data.SqlTypes.SqlBoolean LessThan(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlBoolean LessThanOrEqual(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlMoney Multiply(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlBoolean NotEquals(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlMoney Parse(
   System.String s
);
public static System.Data.SqlTypes.SqlMoney Subtract(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public System.Decimal ToDecimal();
public double ToDouble();
public int ToInt32();
public long ToInt64();
public System.Data.SqlTypes.SqlBoolean ToSqlBoolean();
public System.Data.SqlTypes.SqlByte ToSqlByte();
public System.Data.SqlTypes.SqlDecimal ToSqlDecimal();
public System.Data.SqlTypes.SqlDouble ToSqlDouble();
public System.Data.SqlTypes.SqlInt16 ToSqlInt16();
public System.Data.SqlTypes.SqlInt32 ToSqlInt32();
public System.Data.SqlTypes.SqlInt64 ToSqlInt64();
public System.Data.SqlTypes.SqlSingle ToSqlSingle();
public System.Data.SqlTypes.SqlString ToSqlString();
public System.String ToString();
```

Public Operators

```
public static System.Data.SqlTypes.SqlMoney op_Addition(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlMoney op_Division(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlBoolean op_Equality(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlMoney op_Explicit(
   System.Data.SqlTypes.SqlBoolean x
);
public static System.Data.SqlTypes.SqlMoney op_Explicit(
   System.Data.SqlTypes.SqlDecimal x
);
public static System.Data.SqlTypes.SqlMoney op_Explicit(
   System.Data.SqlTypes.SqlDouble x
);
public static System.Data.SqlTypes.SqlMoney op_Explicit(
   System.Data.SqlTypes.SqlSingle x
);
public static System.Data.SqlTypes.SqlMoney op_Explicit(
   System.Data.SqlTypes.SqlString x
```

```
);
public static System.Decimal op_Explicit(
   System.Data.SqlTypes.SqlMoney x
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThan(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThanOrEqual(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlMoney op_Implicit(
   System.Data.SqlTypes.SqlByte x
);
public static System.Data.SqlTypes.SqlMoney op_Implicit(
   System.Data.SqlTypes.SqlInt16 x
);
public static System.Data.SqlTypes.SqlMoney op_Implicit(
   System.Data.SqlTypes.SqlInt32 x
);
public static System.Data.SqlTypes.SqlMoney op_Implicit(
   System.Data.SqlTypes.SqlInt64 x
);
public static System.Data.SqlTypes.SqlMoney op_Implicit(
   System.Decimal x
);
public static System.Data.SqlTypes.SqlBoolean op_Inequality(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThan(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThanOrEqual(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlMoney op_Multiply(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlMoney op_Subtraction(
   System.Data.SqlTypes.SqlMoney x,
   System.Data.SqlTypes.SqlMoney y
);
public static System.Data.SqlTypes.SqlMoney op_UnaryNegation(
   System.Data.SqlTypes.SqlMoney x
);
```

Public Fields

```
public static final System.Data.SqlTypes.SqlMoney MaxValue;
public static final System.Data.SqlTypes.SqlMoney MinValue;
public static final System.Data.SqlTypes.SqlMoney Null;
public static final System.Data.SqlTypes.SqlMoney Zero;
```

SqlNullValueException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SqlNullValueException
   extends System.Data.SqlTypes.SqlTypeException
```

Public Constructors

```
public SqlNullValueException();
public SqlNullValueException(
   System.String message
);
```

SqlSingle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlSingle
   extends System.ValueType
   implements System.Data.SqlTypes.INullable, System.IComparable
```

Public Constructors

```
public SqlSingle(
   double value
);
public SqlSingle(
   float value
);
```

Public Properties

```
public final boolean get_IsNull();
public float get_Value();
```

Public Methods

```
public static System.Data.SqlTypes.SqlSingle Add(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public final int CompareTo(
   System.Object value
);
public static System.Data.SqlTypes.SqlSingle Divide(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public boolean Equals(
   System.Object value
);
public static System.Data.SqlTypes.SqlBoolean Equals(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public int GetHashCode();
public static System.Data.SqlTypes.SqlBoolean GreaterThan(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlBoolean GreaterThanOrEqual(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlBoolean LessThan(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlBoolean LessThanOrEqual(
   System.Data.SqlTypes.SqlSingle x,
```

```
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlSingle Multiply(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlBoolean NotEquals(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlSingle Parse(
   System.String s
);
public static System.Data.SqlTypes.SqlSingle Subtract(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public System.Data.SqlTypes.SqlBoolean ToSqlBoolean();
public System.Data.SqlTypes.SqlByte ToSqlByte();
public System.Data.SqlTypes.SqlDecimal ToSqlDecimal();
public System.Data.SqlTypes.SqlDouble ToSqlDouble();
public System.Data.SqlTypes.SqlInt16 ToSqlInt16();
public System.Data.SqlTypes.SqlInt32 ToSqlInt32();
public System.Data.SqlTypes.SqlInt64 ToSqlInt64();
public System.Data.SqlTypes.SqlMoney ToSqlMoney();
public System.Data.SqlTypes.SqlString ToSqlString();
public System.String ToString();
```

Public Operators

```
public static System.Data.SqlTypes.SqlSingle op_Addition(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlSingle op_Division(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlBoolean op_Equality(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static float op_Explicit(
   System.Data.SqlTypes.SqlSingle x
);
public static System.Data.SqlTypes.SqlSingle op_Explicit(
   System.Data.SqlTypes.SqlBoolean x
);
public static System.Data.SqlTypes.SqlSingle op_Explicit(
   System.Data.SqlTypes.SqlDouble x
);
public static System.Data.SqlTypes.SqlSingle op_Explicit(
   System.Data.SqlTypes.SqlString x
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThan(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThanOrEqual(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlSingle op_Implicit(
   float x
);
public static System.Data.SqlTypes.SqlSingle op_Implicit(
```

```
   System.Data.SqlTypes.SqlByte x
);
public static System.Data.SqlTypes.SqlSingle op_Implicit(
   System.Data.SqlTypes.SqlDecimal x
);
public static System.Data.SqlTypes.SqlSingle op_Implicit(
   System.Data.SqlTypes.SqlInt16 x
);
public static System.Data.SqlTypes.SqlSingle op_Implicit(
   System.Data.SqlTypes.SqlInt32 x
);
public static System.Data.SqlTypes.SqlSingle op_Implicit(
   System.Data.SqlTypes.SqlInt64 x
);
public static System.Data.SqlTypes.SqlSingle op_Implicit(
   System.Data.SqlTypes.SqlMoney x
);
public static System.Data.SqlTypes.SqlBoolean op_Inequality(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThan(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThanOrEqual(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlSingle op_Multiply(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlSingle op_Subtraction(
   System.Data.SqlTypes.SqlSingle x,
   System.Data.SqlTypes.SqlSingle y
);
public static System.Data.SqlTypes.SqlSingle op_UnaryNegation(
   System.Data.SqlTypes.SqlSingle x
);
```

Public Fields

```
public static final System.Data.SqlTypes.SqlSingle MaxValue;
public static final System.Data.SqlTypes.SqlSingle MinValue;
public static final System.Data.SqlTypes.SqlSingle Null;
public static final System.Data.SqlTypes.SqlSingle Zero;
```

SqlString Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SqlString
   extends System.ValueType
   implements System.Data.SqlTypes.INullable, System.IComparable
```

Public Constructors

```
public SqlString(
   System.String data
);
public SqlString(
   System.String data,
   int lcid
);
public SqlString(
   int lcid,
   System.Data.SqlTypes.SqlCompareOptions compareOptions,
   ubyte[] data
);
public SqlString(
   System.String data,
   int lcid,
   System.Data.SqlTypes.SqlCompareOptions compareOptions
);
public SqlString(
   int lcid,
   System.Data.SqlTypes.SqlCompareOptions compareOptions,
   ubyte[] data,
   boolean fUnicode
);
public SqlString(
   int lcid,
   System.Data.SqlTypes.SqlCompareOptions compareOptions,
   ubyte[] data,
   int index,
   int count
);
public SqlString(
   int lcid,
   System.Data.SqlTypes.SqlCompareOptions compareOptions,
   ubyte[] data,
   int index,
   int count,
   boolean fUnicode
);
```

Public Properties

```
public System.Globalization.CompareInfo get_CompareInfo();
public System.Globalization.CultureInfo get_CultureInfo();
public final boolean get_IsNull();
public int get_LCID();
public System.Data.SqlTypes.SqlCompareOptions get_SqlCompareOptions();
public System.String get_Value();
```

Public Methods

```
public System.Data.SqlTypes.SqlString Clone();
public static System.Globalization.CompareOptions CompareOptionsFromSqlCompareOptions(
   System.Data.SqlTypes.SqlCompareOptions compareOptions
);
public final int CompareTo(
   System.Object value
);
public static System.Data.SqlTypes.SqlString Concat(
   System.Data.SqlTypes.SqlString x,
   System.Data.SqlTypes.SqlString y
);
public boolean Equals(
   System.Object value
);
public static System.Data.SqlTypes.SqlBoolean Equals(
   System.Data.SqlTypes.SqlString x,
   System.Data.SqlTypes.SqlString y
);
public int GetHashCode();
public ubyte[] GetNonUnicodeBytes();
public ubyte[] GetUnicodeBytes();
public static System.Data.SqlTypes.SqlBoolean GreaterThan(
   System.Data.SqlTypes.SqlString x,
   System.Data.SqlTypes.SqlString y
);
public static System.Data.SqlTypes.SqlBoolean GreaterThanOrEqual(
   System.Data.SqlTypes.SqlString x,
   System.Data.SqlTypes.SqlString y
);
public static System.Data.SqlTypes.SqlBoolean LessThan(
   System.Data.SqlTypes.SqlString x,
   System.Data.SqlTypes.SqlString y
);
public static System.Data.SqlTypes.SqlBoolean LessThanOrEqual(
   System.Data.SqlTypes.SqlString x,
   System.Data.SqlTypes.SqlString y
);
public static System.Data.SqlTypes.SqlBoolean NotEquals(
   System.Data.SqlTypes.SqlString x,
   System.Data.SqlTypes.SqlString y
);
public System.Data.SqlTypes.SqlBoolean ToSqlBoolean();
public System.Data.SqlTypes.SqlByte ToSqlByte();
public System.Data.SqlTypes.SqlDateTime ToSqlDateTime();
public System.Data.SqlTypes.SqlDecimal ToSqlDecimal();
public System.Data.SqlTypes.SqlDouble ToSqlDouble();
public System.Data.SqlTypes.SqlGuid ToSqlGuid();
public System.Data.SqlTypes.SqlInt16 ToSqlInt16();
public System.Data.SqlTypes.SqlInt32 ToSqlInt32();
public System.Data.SqlTypes.SqlInt64 ToSqlInt64();
public System.Data.SqlTypes.SqlMoney ToSqlMoney();
public System.Data.SqlTypes.SqlSingle ToSqlSingle();
public System.String ToString();
```

Public Operators

```
public static System.Data.SqlTypes.SqlString op_Addition(
   System.Data.SqlTypes.SqlString x,
   System.Data.SqlTypes.SqlString y
);
public static System.Data.SqlTypes.SqlBoolean op_Equality(
   System.Data.SqlTypes.SqlString x,
   System.Data.SqlTypes.SqlString y
```

```
);
public static System.Data.SqlTypes.SqlString op_Explicit(
   System.Data.SqlTypes.SqlBoolean x
);
public static System.Data.SqlTypes.SqlString op_Explicit(
   System.Data.SqlTypes.SqlByte x
);
public static System.Data.SqlTypes.SqlString op_Explicit(
   System.Data.SqlTypes.SqlDateTime x
);
public static System.Data.SqlTypes.SqlString op_Explicit(
   System.Data.SqlTypes.SqlDecimal x
);
public static System.Data.SqlTypes.SqlString op_Explicit(
   System.Data.SqlTypes.SqlDouble x
);
public static System.Data.SqlTypes.SqlString op_Explicit(
   System.Data.SqlTypes.SqlGuid x
);
public static System.Data.SqlTypes.SqlString op_Explicit(
   System.Data.SqlTypes.SqlInt16 x
);
public static System.Data.SqlTypes.SqlString op_Explicit(
   System.Data.SqlTypes.SqlInt32 x
);
public static System.Data.SqlTypes.SqlString op_Explicit(
   System.Data.SqlTypes.SqlInt64 x
);
public static System.Data.SqlTypes.SqlString op_Explicit(
   System.Data.SqlTypes.SqlMoney x
);
public static System.Data.SqlTypes.SqlString op_Explicit(
   System.Data.SqlTypes.SqlSingle x
);
public static System.String op_Explicit(
   System.Data.SqlTypes.SqlString x
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThan(
   System.Data.SqlTypes.SqlString x,
   System.Data.SqlTypes.SqlString y
);
public static System.Data.SqlTypes.SqlBoolean op_GreaterThanOrEqual(
   System.Data.SqlTypes.SqlString x,
   System.Data.SqlTypes.SqlString y
);
public static System.Data.SqlTypes.SqlString op_Implicit(
   System.String x
);
public static System.Data.SqlTypes.SqlBoolean op_Inequality(
   System.Data.SqlTypes.SqlString x,
   System.Data.SqlTypes.SqlString y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThan(
   System.Data.SqlTypes.SqlString x,
   System.Data.SqlTypes.SqlString y
);
public static System.Data.SqlTypes.SqlBoolean op_LessThanOrEqual(
   System.Data.SqlTypes.SqlString x,
   System.Data.SqlTypes.SqlString y
);
```

Public Fields

```
public static final int BinarySort;
public static final int IgnoreCase;
public static final int IgnoreKanaType;
```

```
public static final int IgnoreNonSpace;
public static final int IgnoreWidth;
public static final System.Data.SqlTypes.SqlString Null;
```

SqlTruncateException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SqlTruncateException
   extends System.Data.SqlTypes.SqlTypeException
```

Public Constructors

```
public SqlTruncateException();
public SqlTruncateException(
   System.String message
);
```

SqlTypeException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SqlTypeException
   extends System.SystemException
```

Public Constructors

```
public SqlTypeException(
   System.String message
);
```

Protected Constructors

```
protected SqlTypeException(
   System.Runtime.Serialization.SerializationInfo si,
   System.Runtime.Serialization.StreamingContext sc
);
```

System.Diagnostics Namespace (Visual J# Syntax)

Classes

BooleanSwitch

ConditionalAttribute

CounterCreationData

CounterCreationDataCollection

CounterSampleCalculator

Debug

DebuggableAttribute

Debugger

DebuggerHiddenAttribute

DebuggerStepThroughAttribute

DefaultTraceListener

DiagnosticsConfigurationHandler

EntryWrittenEventArgs

EventLog

EventLogEntry

EventLogEntryCollection

EventLogInstaller

EventLogPermission

EventLogPermissionAttribute

EventLogPermissionEntry

EventLogPermissionEntryCollection

EventLogTraceListener

FileVersionInfo

InstanceData

InstanceDataCollection

InstanceDataCollectionCollection

MonitoringDescriptionAttribute

PerformanceCounter

PerformanceCounterCategory

PerformanceCounterInstaller

PerformanceCounterManager

PerformanceCounterPermission

PerformanceCounterPermissionAttribute

PerformanceCounterPermissionEntry

PerformanceCounterPermissionEntryCollection

Process

ProcessModule

ProcessModuleCollection

ProcessStartInfo

ProcessThread

ProcessThreadCollection

StackFrame

StackTrace

Switch

TextWriterTraceListener

Trace

TraceListener

TraceListenerCollection

TraceSwitch

Interfaces

ICollectData

Structures

CounterSample

Delegates

EntryWrittenEventHandler

Enumerations

EventLogEntryType

EventLogPermissionAccess

PerformanceCounterPermissionAccess

PerformanceCounterType

ProcessPriorityClass

ProcessWindowStyle

ThreadPriorityLevel

ThreadState

ThreadWaitReason

TraceLevel

BooleanSwitch Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BooleanSwitch
   extends System.Diagnostics.Switch
```

Public Constructors

```
public BooleanSwitch(
   System.String displayName,
   System.String description
);
```

Public Properties

```
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
```

ConditionalAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = true, Inherit
ed = true) */
public class ConditionalAttribute
   extends System.Attribute
```

Public Constructors

```
public ConditionalAttribute(
   System.String conditionString
);
```

Public Properties

```
public System.String get_ConditionString();
```

CounterCreationData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Diagn
ostics.Design.CounterCreationDataConverter, System.Design, Version=1.0.3300.0, Culture=neutra
l, PublicKeyToken=b03f5f7f11d50a3a") */
public class CounterCreationData
   extends System.Object
```

Public Constructors

```
public CounterCreationData();
public CounterCreationData(
   System.String counterName,
   System.String counterHelp,
   System.Diagnostics.PerformanceCounterType counterType
);
```

Public Properties

```
public System.String get_CounterHelp();
public void set_CounterHelp(
   System.String value
);
public System.String get_CounterName();
public void set_CounterName(
   System.String value
);
public System.Diagnostics.PerformanceCounterType get_CounterType();
public void set_CounterType(
   System.Diagnostics.PerformanceCounterType value
);
```

CounterCreationDataCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CounterCreationDataCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public CounterCreationDataCollection();
public CounterCreationDataCollection(
   System.Diagnostics.CounterCreationDataCollection value
);
public CounterCreationDataCollection(
   System.Diagnostics.CounterCreationData[] value
);
```

Public Properties

```
public System.Diagnostics.CounterCreationData get_Item(
   int index
);
public void set_Item(
   int index,
   System.Diagnostics.CounterCreationData value
);
```

Public Methods

```
public int Add(
   System.Diagnostics.CounterCreationData value
);
public void AddRange(
   System.Diagnostics.CounterCreationDataCollection value
);
public void AddRange(
   System.Diagnostics.CounterCreationData[] value
);
public boolean Contains(
   System.Diagnostics.CounterCreationData value
);
public void CopyTo(
   System.Diagnostics.CounterCreationData[] array,
   int index
);
public int IndexOf(
   System.Diagnostics.CounterCreationData value
);
public void Insert(
   int index,
   System.Diagnostics.CounterCreationData value
);
public void Remove(
   System.Diagnostics.CounterCreationData value
);
```

Protected Methods

```
protected void OnInsert(
   int index,
   System.Object value
);
```

CounterSample Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CounterSample
   extends System.ValueType
```

Public Constructors

```
public CounterSample(
   long rawValue,
   long baseValue,
   long counterFrequency,
   long systemFrequency,
   long timeStamp,
   long timeStamp100nSec,
   System.Diagnostics.PerformanceCounterType counterType
);
public CounterSample(
   long rawValue,
   long baseValue,
   long counterFrequency,
   long systemFrequency,
   long timeStamp,
   long timeStamp100nSec,
   System.Diagnostics.PerformanceCounterType counterType,
   long counterTimeStamp
);
```

Public Properties

```
public long get_BaseValue();
public long get_CounterFrequency();
public long get_CounterTimeStamp();
public System.Diagnostics.PerformanceCounterType get_CounterType();
public long get_RawValue();
public long get_SystemFrequency();
public long get_TimeStamp();
public long get_TimeStamp100nSec();
```

Public Methods

```
public static float Calculate(
   System.Diagnostics.CounterSample counterSample
);
public static float Calculate(
   System.Diagnostics.CounterSample counterSample,
   System.Diagnostics.CounterSample nextCounterSample
);
```

Public Fields

```
public static System.Diagnostics.CounterSample Empty;
```

CounterSampleCalculator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CounterSampleCalculator
   extends System.Object
```

Public Methods

```
public static float ComputeCounterValue(
   System.Diagnostics.CounterSample newSample
);
public static float ComputeCounterValue(
   System.Diagnostics.CounterSample oldSample,
   System.Diagnostics.CounterSample newSample
);
```

Debug Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Debug
   extends System.Object
```

Public Properties

```
public static boolean get_AutoFlush();
public static void set_AutoFlush(
   boolean value
);
public static int get_IndentLevel();
public static void set_IndentLevel(
   int value
);
public static int get_IndentSize();
public static void set_IndentSize(
   int value
);
public static System.Diagnostics.TraceListenerCollection get_Listeners();
```

Public Methods

```
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void Assert(
   boolean condition
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void Assert(
   boolean condition,
   System.String message
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void Assert(
   boolean condition,
   System.String message,
   System.String detailMessage
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void Close();
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void Fail(
   System.String message
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void Fail(
   System.String message,
   System.String detailMessage
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void Flush();
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void Indent();
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void Unindent();
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void Write(
```

```
   System.Object value
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void Write(
   System.String message
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void Write(
   System.Object value,
   System.String category
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void Write(
   System.String message,
   System.String category
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void WriteIf(
   boolean condition,
   System.Object value
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void WriteIf(
   boolean condition,
   System.String message
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void WriteIf(
   boolean condition,
   System.Object value,
   System.String category
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void WriteIf(
   boolean condition,
   System.String message,
   System.String category
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void WriteLine(
   System.Object value
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void WriteLine(
   System.String message
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void WriteLine(
   System.Object value,
   System.String category
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void WriteLine(
   System.String message,
   System.String category
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void WriteLineIf(
   boolean condition,
   System.Object value
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void WriteLineIf(
   boolean condition,
   System.String message
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void WriteLineIf(
```

```
   boolean condition,
   System.Object value,
   System.String category
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "DEBUG") */
public static void WriteLineIf(
   boolean condition,
   System.String message,
   System.String category
);
```

DebuggableAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Module, AllowMultiple = fal
se, Inherited = true) */
public class DebuggableAttribute
   extends System.Attribute
```

Public Constructors

```
public DebuggableAttribute(
   boolean isJITTrackingEnabled,
   boolean isJITOptimizerDisabled
);
```

Public Properties

```
public boolean get_IsJITOptimizerDisabled();
public boolean get_IsJITTrackingEnabled();
```

Debugger Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Debugger
   extends System.Object
```

Public Constructors

```
public Debugger();
```

Public Properties

```
public static boolean get_IsAttached();
```

Public Methods

```
public static void Break();
public static boolean IsLogging();
public static boolean Launch();
public static void Log(
   int level,
   System.String category,
   System.String message
);
```

Public Fields

```
public static final System.String DefaultCategory;
```

DebuggerHiddenAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Constructor, Method, Property, AllowM
ultiple = false, Inherited = false) */
public class DebuggerHiddenAttribute
   extends System.Attribute
```

Public Constructors

```
public DebuggerHiddenAttribute();
```

DebuggerStepThroughAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, Constructor, Method, A
llowMultiple = false, Inherited = false) */
public class DebuggerStepThroughAttribute
   extends System.Attribute
```

Public Constructors

```
public DebuggerStepThroughAttribute();
```

DefaultTraceListener Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class DefaultTraceListener
   extends System.Diagnostics.TraceListener
```

Public Constructors

```
public DefaultTraceListener();
```

Public Properties

```
public boolean get_AssertUiEnabled();
public void set_AssertUiEnabled(
   boolean value
);
public System.String get_LogFileName();
public void set_LogFileName(
   System.String value
);
```

Public Methods

```
public void Fail(
   System.String message
);
public void Fail(
   System.String message,
   System.String detailMessage
);
public void Write(
   System.String message
);
public void WriteLine(
   System.String message
);
```

DiagnosticsConfigurationHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DiagnosticsConfigurationHandler
   extends System.Object
   implements System.Configuration.IConfigurationSectionHandler
```

Public Constructors

```
public DiagnosticsConfigurationHandler();
```

Public Methods

```
public System.Object Create(
   System.Object parent,
   System.Object configContext,
   System.Xml.XmlNode section
);
```

EntryWrittenEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EntryWrittenEventArgs
   extends System.EventArgs
```

Public Constructors

```
public EntryWrittenEventArgs();
public EntryWrittenEventArgs(
   System.Diagnostics.EventLogEntry entry
);
```

Public Properties

```
public System.Diagnostics.EventLogEntry get_Entry();
```

EntryWrittenEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void EntryWrittenEventHandler(
   System.Object sender,
   System.Diagnostics.EntryWrittenEventArgs e
);
```

EventLog Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "Microsoft.VisualStudio.Install.EventLogInstall
ableComponentDesigner, Microsoft.VisualStudio, Version=1.0.3300.0, Culture=neutral, PublicKey
Token=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "EntryWritten") */
/** @attribute System.ComponentModel.InstallerTypeAttribute(InstallerType = System.Diagnostic
s.EventLogInstaller) */
public class EventLog
   extends System.ComponentModel.Component
   implements System.ComponentModel.ISupportInitialize
```

Public Constructors

```
public EventLog();
public EventLog(
   System.String logName
);
public EventLog(
   System.String logName,
   System.String machineName
);
public EventLog(
   System.String logName,
   System.String machineName,
   System.String source
);
```

Public Properties

```
public boolean get_EnableRaisingEvents();
public void set_EnableRaisingEvents(
   boolean value
);
public System.Diagnostics.EventLogEntryCollection get_Entries();
public System.String get_Log();
public void set_Log(
   System.String value
);
public System.String get_LogDisplayName();
public System.String get_MachineName();
public void set_MachineName(
   System.String value
);
public System.String get_Source();
public void set_Source(
   System.String value
);
public System.ComponentModel.ISynchronizeInvoke get_SynchronizingObject();
public void set_SynchronizingObject(
   System.ComponentModel.ISynchronizeInvoke value
);
```

Public Methods

```
public final void BeginInit();
public void Clear();
public void Close();
public static void CreateEventSource(
   System.String source,
   System.String logName
);
public static void CreateEventSource(
   System.String source,
   System.String logName,
   System.String machineName
);
public static void Delete(
   System.String logName
);
public static void Delete(
   System.String logName,
   System.String machineName
);
public static void DeleteEventSource(
   System.String source
);
public static void DeleteEventSource(
   System.String source,
   System.String machineName
);
public final void EndInit();
public static boolean Exists(
   System.String logName
);
public static boolean Exists(
   System.String logName,
   System.String machineName
);
public static System.Diagnostics.EventLog[] GetEventLogs();
public static System.Diagnostics.EventLog[] GetEventLogs(
   System.String machineName
);
public static System.String LogNameFromSourceName(
   System.String source,
   System.String machineName
);
public static boolean SourceExists(
   System.String source
);
public static boolean SourceExists(
   System.String source,
   System.String machineName
);
public void WriteEntry(
   System.String message
);
public void WriteEntry(
   System.String message,
   System.Diagnostics.EventLogEntryType type
);
public static void WriteEntry(
   System.String source,
   System.String message
);
public void WriteEntry(
   System.String message,
   System.Diagnostics.EventLogEntryType type,
   int eventID
);
public static void WriteEntry(
   System.String source,
   System.String message,
```

```
   System.Diagnostics.EventLogEntryType type
);
public void WriteEntry(
   System.String message,
   System.Diagnostics.EventLogEntryType type,
   int eventID,
   short category
);
public static void WriteEntry(
   System.String source,
   System.String message,
   System.Diagnostics.EventLogEntryType type,
   int eventID
);
public void WriteEntry(
   System.String message,
   System.Diagnostics.EventLogEntryType type,
   int eventID,
   short category,
   ubyte[] rawData
);
public static void WriteEntry(
   System.String source,
   System.String message,
   System.Diagnostics.EventLogEntryType type,
   int eventID,
   short category
);
public static void WriteEntry(
   System.String source,
   System.String message,
   System.Diagnostics.EventLogEntryType type,
   int eventID,
   short category,
   ubyte[] rawData
);
```

Public Events

```
public void add_EntryWritten(
   System.Diagnostics.EntryWrittenEventHandler value
);
public void remove_EntryWritten(
   System.Diagnostics.EntryWrittenEventHandler value
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

EventLogEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
public class EventLogEntry
   extends System.ComponentModel.Component
   implements System.Runtime.Serialization.ISerializable
```

Public Properties

```
public System.String get_Category();
public short get_CategoryNumber();
public ubyte[] get_Data();
public System.Diagnostics.EventLogEntryType get_EntryType();
public int get_EventID();
public int get_Index();
public System.String get_MachineName();
public System.String get_Message();
public System.String[] get_ReplacementStrings();
public System.String get_Source();
public System.DateTime get_TimeGenerated();
public System.DateTime get_TimeWritten();
public System.String get_UserName();
```

Public Methods

```
public boolean Equals(
   System.Diagnostics.EventLogEntry otherEntry
);
```

EventLogEntryCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class EventLogEntryCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public System.Diagnostics.EventLogEntry get_Item(
   int index
);
```

Public Methods

```
public void CopyTo(
   System.Diagnostics.EventLogEntry[] entries,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
```

EventLogEntryType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum EventLogEntryType
```

Members

Error

FailureAudit

Information

SuccessAudit

Warning

EventLogInstaller Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EventLogInstaller
   extends System.Configuration.Install.ComponentInstaller
```

Public Constructors

```
public EventLogInstaller();
```

Public Properties

```
public System.String get_Log();
public void set_Log(
   System.String value
);
public System.String get_Source();
public void set_Source(
   System.String value
);
public System.Configuration.Install.UninstallAction get_UninstallAction();
public void set_UninstallAction(
   System.Configuration.Install.UninstallAction value
);
```

Public Methods

```
public void CopyFromComponent(
   System.ComponentModel.IComponent component
);
public void Install(
   System.Collections.IDictionary stateSaver
);
public boolean IsEquivalentInstaller(
   System.Configuration.Install.ComponentInstaller otherInstaller
);
public void Rollback(
   System.Collections.IDictionary savedState
);
public void Uninstall(
   System.Collections.IDictionary savedState
);
```

EventLogPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class EventLogPermission
   extends System.Security.Permissions.ResourcePermissionBase
```

Public Constructors

```
public EventLogPermission();
public EventLogPermission(
   System.Diagnostics.EventLogPermissionEntry[] permissionAccessEntries
);
public EventLogPermission(
   System.Security.Permissions.PermissionState state
);
public EventLogPermission(
   System.Diagnostics.EventLogPermissionAccess permissionAccess,
   System.String machineName
);
```

Public Properties

```
public System.Diagnostics.EventLogPermissionEntryCollection get_PermissionEntries();
```

EventLogPermissionAccess Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum EventLogPermissionAccess
```

Members

Audit

Browse

Instrument

None

EventLogPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, Event, AllowMultiple = true, Inherited = false) */
public class EventLogPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public EventLogPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.String get_MachineName();
public void set_MachineName(
   System.String value
);
public System.Diagnostics.EventLogPermissionAccess get_PermissionAccess();
public void set_PermissionAccess(
   System.Diagnostics.EventLogPermissionAccess value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

EventLogPermissionEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class EventLogPermissionEntry
   extends System.Object
```

Public Constructors

```
public EventLogPermissionEntry(
   System.Diagnostics.EventLogPermissionAccess permissionAccess,
   System.String machineName
);
```

Public Properties

```
public System.String get_MachineName();
public System.Diagnostics.EventLogPermissionAccess get_PermissionAccess();
```

EventLogPermissionEntryCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class EventLogPermissionEntryCollection
   extends System.Collections.CollectionBase
```

Public Properties

```
public System.Diagnostics.EventLogPermissionEntry get_Item(
   int index
);
public void set_Item(
   int index,
   System.Diagnostics.EventLogPermissionEntry value
);
```

Public Methods

```
public int Add(
   System.Diagnostics.EventLogPermissionEntry value
);
public void AddRange(
   System.Diagnostics.EventLogPermissionEntryCollection value
);
public void AddRange(
   System.Diagnostics.EventLogPermissionEntry[] value
);
public boolean Contains(
   System.Diagnostics.EventLogPermissionEntry value
);
public void CopyTo(
   System.Diagnostics.EventLogPermissionEntry[] array,
   int index
);
public int IndexOf(
   System.Diagnostics.EventLogPermissionEntry value
);
public void Insert(
   int index,
   System.Diagnostics.EventLogPermissionEntry value
);
public void Remove(
   System.Diagnostics.EventLogPermissionEntry value
);
```

Protected Methods

```
protected void OnClear();
protected void OnInsert(
   int index,
   System.Object value
```

```
);
protected void OnRemove(
   int index,
   System.Object value
);
protected void OnSet(
   int index,
   System.Object oldValue,
   System.Object newValue
);
```

EventLogTraceListener Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EventLogTraceListener
   extends System.Diagnostics.TraceListener
```

Public Constructors

```
public EventLogTraceListener();
public EventLogTraceListener(
   System.Diagnostics.EventLog eventLog
);
public EventLogTraceListener(
   System.String source
);
```

Public Properties

```
public System.Diagnostics.EventLog get_EventLog();
public void set_EventLog(
   System.Diagnostics.EventLog value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
```

Public Methods

```
public void Close();
public void Write(
   System.String message
);
public void WriteLine(
   System.String message
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

FileVersionInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FileVersionInfo
   extends System.Object
```

Public Properties

```
public System.String get_Comments();
public System.String get_CompanyName();
public int get_FileBuildPart();
public System.String get_FileDescription();
public int get_FileMajorPart();
public int get_FileMinorPart();
public System.String get_FileName();
public int get_FilePrivatePart();
public System.String get_FileVersion();
public System.String get_InternalName();
public boolean get_IsDebug();
public boolean get_IsPatched();
public boolean get_IsPreRelease();
public boolean get_IsPrivateBuild();
public boolean get_IsSpecialBuild();
public System.String get_Language();
public System.String get_LegalCopyright();
public System.String get_LegalTrademarks();
public System.String get_OriginalFilename();
public System.String get_PrivateBuild();
public int get_ProductBuildPart();
public int get_ProductMajorPart();
public int get_ProductMinorPart();
public System.String get_ProductName();
public int get_ProductPrivatePart();
public System.String get_ProductVersion();
public System.String get_SpecialBuild();
```

Public Methods

```
public static System.Diagnostics.FileVersionInfo GetVersionInfo(
   System.String fileName
);
public System.String ToString();
```

ICollectData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "73386977-D6FD-11D2-BED5-
00C04F79E3AE") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface ICollectData
```

Public Methods

```
void CloseData();
void CollectData(
   int id,
   System.IntPtr valueName,
   System.IntPtr data,
   int totalBytes,
   System.IntPtr& res
);
```

InstanceData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InstanceData
   extends System.Object
```

Public Constructors

```
public InstanceData(
   System.String instanceName,
   System.Diagnostics.CounterSample sample
);
```

Public Properties

```
public System.String get_InstanceName();
public long get_RawValue();
public System.Diagnostics.CounterSample get_Sample();
```

InstanceDataCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class InstanceDataCollection
   extends System.Collections.DictionaryBase
```

Public Constructors

```
public InstanceDataCollection(
   System.String counterName
);
```

Public Properties

```
public System.String get_CounterName();
public System.Diagnostics.InstanceData get_Item(
   System.String instanceName
);
public System.Collections.ICollection get_Keys();
public System.Collections.ICollection get_Values();
```

Public Methods

```
public boolean Contains(
   System.String instanceName
);
public void CopyTo(
   System.Diagnostics.InstanceData[] instances,
   int index
);
```

InstanceDataCollectionCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class InstanceDataCollectionCollection
   extends System.Collections.DictionaryBase
```

Public Constructors

```
public InstanceDataCollectionCollection();
```

Public Properties

```
public System.Diagnostics.InstanceDataCollection get_Item(
   System.String counterName
);
public System.Collections.ICollection get_Keys();
public System.Collections.ICollection get_Values();
```

Public Methods

```
public boolean Contains(
   System.String counterName
);
public void CopyTo(
   System.Diagnostics.InstanceDataCollection[] counters,
   int index
);
```

MonitoringDescriptionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class MonitoringDescriptionAttribute
   extends System.ComponentModel.DescriptionAttribute
```

Public Constructors

```
public MonitoringDescriptionAttribute(
   System.String description
);
```

Public Properties

```
public System.String get_Description();
```

PerformanceCounter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "Microsoft.VisualStudio.Install.PerformanceCoun
terDesigner, Microsoft.VisualStudio, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f
5f7f11d50a3a") */
/** @attribute System.ComponentModel.InstallerTypeAttribute(InstallerType = System.Diagnostic
s.PerformanceCounterInstaller) */
public class PerformanceCounter
   extends System.ComponentModel.Component
   implements System.ComponentModel.ISupportInitialize
```

Public Constructors

```
public PerformanceCounter();
public PerformanceCounter(
   System.String categoryName,
   System.String counterName
);
public PerformanceCounter(
   System.String categoryName,
   System.String counterName,
   boolean readOnly
);
public PerformanceCounter(
   System.String categoryName,
   System.String counterName,
   System.String instanceName
);
public PerformanceCounter(
   System.String categoryName,
   System.String counterName,
   System.String instanceName,
   boolean readOnly
);
public PerformanceCounter(
   System.String categoryName,
   System.String counterName,
   System.String instanceName,
   System.String machineName
);
```

Public Properties

```
public System.String get_CategoryName();
public void set_CategoryName(
   System.String value
);
public System.String get_CounterHelp();
public System.String get_CounterName();
public void set_CounterName(
   System.String value
);
public System.Diagnostics.PerformanceCounterType get_CounterType();
public System.String get_InstanceName();
public void set_InstanceName(
   System.String value
```

```
);
public System.String get_MachineName();
public void set_MachineName(
   System.String value
);
public long get_RawValue();
public void set_RawValue(
   long value
);
public boolean get_ReadOnly();
public void set_ReadOnly(
   boolean value
);
```

Public Methods

```
public final void BeginInit();
public void Close();
public static void CloseSharedResources();
public long Decrement();
public final void EndInit();
public long Increment();
public long IncrementBy(
   long value
);
public System.Diagnostics.CounterSample NextSample();
public float NextValue();
public void RemoveInstance();
```

Public Fields

```
public static int DefaultFileMappingSize;
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

PerformanceCounterCategory Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PerformanceCounterCategory
   extends System.Object
```

Public Constructors

```
public PerformanceCounterCategory();
public PerformanceCounterCategory(
   System.String categoryName
);
public PerformanceCounterCategory(
   System.String categoryName,
   System.String machineName
);
```

Public Properties

```
public System.String get_CategoryHelp();
public System.String get_CategoryName();
public void set_CategoryName(
   System.String value
);
public System.String get_MachineName();
public void set_MachineName(
   System.String value
);
```

Public Methods

```
public boolean CounterExists(
   System.String counterName
);
public static boolean CounterExists(
   System.String counterName,
   System.String categoryName
);
public static boolean CounterExists(
   System.String counterName,
   System.String categoryName,
   System.String machineName
);
public static System.Diagnostics.PerformanceCounterCategory Create(
   System.String categoryName,
   System.String categoryHelp,
   System.Diagnostics.CounterCreationDataCollection counterData
);
public static System.Diagnostics.PerformanceCounterCategory Create(
   System.String categoryName,
   System.String categoryHelp,
   System.String counterName,
   System.String counterHelp
);
public static void Delete(
   System.String categoryName
```

```
);
public static boolean Exists(
   System.String categoryName
);
public static boolean Exists(
   System.String categoryName,
   System.String machineName
);
public static System.Diagnostics.PerformanceCounterCategory[] GetCategories();
public static System.Diagnostics.PerformanceCounterCategory[] GetCategories(
   System.String machineName
);
public System.Diagnostics.PerformanceCounter[] GetCounters();
public System.Diagnostics.PerformanceCounter[] GetCounters(
   System.String instanceName
);
public System.String[] GetInstanceNames();
public boolean InstanceExists(
   System.String instanceName
);
public static boolean InstanceExists(
   System.String instanceName,
   System.String categoryName
);
public static boolean InstanceExists(
   System.String instanceName,
   System.String categoryName,
   System.String machineName
);
public System.Diagnostics.InstanceDataCollectionCollection ReadCategory();
```

PerformanceCounterInstaller Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PerformanceCounterInstaller
   extends System.Configuration.Install.ComponentInstaller
```

Public Constructors

```
public PerformanceCounterInstaller();
```

Public Properties

```
public System.String get_CategoryHelp();
public void set_CategoryHelp(
   System.String value
);
public System.String get_CategoryName();
public void set_CategoryName(
   System.String value
);
public System.Diagnostics.CounterCreationDataCollection get_Counters();
public System.Configuration.Install.UninstallAction get_UninstallAction();
public void set_UninstallAction(
   System.Configuration.Install.UninstallAction value
);
```

Public Methods

```
public void CopyFromComponent(
   System.ComponentModel.IComponent component
);
public void Install(
   System.Collections.IDictionary stateSaver
);
public void Rollback(
   System.Collections.IDictionary savedState
);
public void Uninstall(
   System.Collections.IDictionary savedState
);
```

PerformanceCounterManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "82840BE1-D273-11D2-B94A-
00600893B17A") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class PerformanceCounterManager
   extends System.Object
   implements System.Diagnostics.ICollectData
```

Public Constructors

```
public PerformanceCounterManager();
```

PerformanceCounterPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class PerformanceCounterPermission
   extends System.Security.Permissions.ResourcePermissionBase
```

Public Constructors

```
public PerformanceCounterPermission();
public PerformanceCounterPermission(
   System.Diagnostics.PerformanceCounterPermissionEntry[] permissionAccessEntries
);
public PerformanceCounterPermission(
   System.Security.Permissions.PermissionState state
);
public PerformanceCounterPermission(
   System.Diagnostics.PerformanceCounterPermissionAccess permissionAccess,
   System.String machineName,
   System.String categoryName
);
```

Public Properties

```
public System.Diagnostics.PerformanceCounterPermissionEntryCollection get_PermissionEntries()
;
```

PerformanceCounterPermissionAccess Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum PerformanceCounterPermissionAccess
```

Members

Administer

Browse

Instrument

None

PerformanceCounterPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, Event, AllowMultiple = true, Inherited = false) */
public class PerformanceCounterPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public PerformanceCounterPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.String get_CategoryName();
public void set_CategoryName(
   System.String value
);
public System.String get_MachineName();
public void set_MachineName(
   System.String value
);
public System.Diagnostics.PerformanceCounterPermissionAccess get_PermissionAccess();
public void set_PermissionAccess(
   System.Diagnostics.PerformanceCounterPermissionAccess value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

PerformanceCounterPermissionEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class PerformanceCounterPermissionEntry
   extends System.Object
```

Public Constructors

```
public PerformanceCounterPermissionEntry(
   System.Diagnostics.PerformanceCounterPermissionAccess permissionAccess,
   System.String machineName,
   System.String categoryName
);
```

Public Properties

```
public System.String get_CategoryName();
public System.String get_MachineName();
public System.Diagnostics.PerformanceCounterPermissionAccess get_PermissionAccess();
```

PerformanceCounterPermissionEntryCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class PerformanceCounterPermissionEntryCollection
   extends System.Collections.CollectionBase
```

Public Properties

```
public System.Diagnostics.PerformanceCounterPermissionEntry get_Item(
   int index
);
public void set_Item(
   int index,
   System.Diagnostics.PerformanceCounterPermissionEntry value
);
```

Public Methods

```
public int Add(
   System.Diagnostics.PerformanceCounterPermissionEntry value
);
public void AddRange(
   System.Diagnostics.PerformanceCounterPermissionEntryCollection value
);
public void AddRange(
   System.Diagnostics.PerformanceCounterPermissionEntry[] value
);
public boolean Contains(
   System.Diagnostics.PerformanceCounterPermissionEntry value
);
public void CopyTo(
   System.Diagnostics.PerformanceCounterPermissionEntry[] array,
   int index
);
public int IndexOf(
   System.Diagnostics.PerformanceCounterPermissionEntry value
);
public void Insert(
   int index,
   System.Diagnostics.PerformanceCounterPermissionEntry value
);
public void Remove(
   System.Diagnostics.PerformanceCounterPermissionEntry value
);
```

Protected Methods

```
protected void OnClear();
protected void OnInsert(
   int index,
   System.Object value
```

```
);
protected void OnRemove(
   int index,
   System.Object value
);
protected void OnSet(
   int index,
   System.Object oldValue,
   System.Object newValue
);
```

PerformanceCounterType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Diagn
ostics.AlphabeticalEnumConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken
=b77a5c561934e089") */
public enum PerformanceCounterType
```

Members

AverageBase

AverageCount64

AverageTimer32

CounterDelta32

CounterDelta64

CounterMultiBase

CounterMultiTimer

CounterMultiTimer100Ns

CounterMultiTimer100NsInverse

CounterMultiTimerInverse

CounterTimer

CounterTimerInverse

CountPerTimeInterval32

CountPerTimeInterval64

ElapsedTime

NumberOfItems32

NumberOfItems64

NumberOfItemsHEX32

NumberOfItemsHEX64

RateOfCountsPerSecond32

RateOfCountsPerSecond64

RawBase

RawFraction

SampleBase

SampleCounter

SampleFraction

Timer100Ns

Timer100NsInverse

Process Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "StartInfo") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Diagnostics.Design.ProcessDesigner, Sys
tem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Exited") */
public class Process
   extends System.ComponentModel.Component
```

Public Constructors

```
public Process();
```

Public Properties

```
public int get_BasePriority();
public boolean get_EnableRaisingEvents();
public void set_EnableRaisingEvents(
   boolean value
);
public int get_ExitCode();
public System.DateTime get_ExitTime();
public System.IntPtr get_Handle();
public int get_HandleCount();
public boolean get_HasExited();
public int get_Id();
public System.String get_MachineName();
public System.Diagnostics.ProcessModule get_MainModule();
public System.IntPtr get_MainWindowHandle();
public System.String get_MainWindowTitle();
public System.IntPtr get_MaxWorkingSet();
public void set_MaxWorkingSet(
   System.IntPtr value
);
public System.IntPtr get_MinWorkingSet();
public void set_MinWorkingSet(
   System.IntPtr value
);
public System.Diagnostics.ProcessModuleCollection get_Modules();
public int get_NonpagedSystemMemorySize();
public int get_PagedMemorySize();
public int get_PagedSystemMemorySize();
public int get_PeakPagedMemorySize();
public int get_PeakVirtualMemorySize();
public int get_PeakWorkingSet();
public boolean get_PriorityBoostEnabled();
public void set_PriorityBoostEnabled(
   boolean value
);
public System.Diagnostics.ProcessPriorityClass get_PriorityClass();
public void set_PriorityClass(
   System.Diagnostics.ProcessPriorityClass value
);
public int get_PrivateMemorySize();
public System.TimeSpan get_PrivilegedProcessorTime();
public System.String get_ProcessName();
```

```
public System.IntPtr get_ProcessorAffinity();
public void set_ProcessorAffinity(
   System.IntPtr value
);
public boolean get_Responding();
public System.IO.StreamReader get_StandardError();
public System.IO.StreamWriter get_StandardInput();
public System.IO.StreamReader get_StandardOutput();
public System.Diagnostics.ProcessStartInfo get_StartInfo();
public void set_StartInfo(
   System.Diagnostics.ProcessStartInfo value
);
public System.DateTime get_StartTime();
public System.ComponentModel.ISynchronizeInvoke get_SynchronizingObject();
public void set_SynchronizingObject(
   System.ComponentModel.ISynchronizeInvoke value
);
public System.Diagnostics.ProcessThreadCollection get_Threads();
public System.TimeSpan get_TotalProcessorTime();
public System.TimeSpan get_UserProcessorTime();
public int get_VirtualMemorySize();
public int get_WorkingSet();
```

Public Methods

```
public void Close();
public boolean CloseMainWindow();
public static void EnterDebugMode();
public static System.Diagnostics.Process GetCurrentProcess();
public static System.Diagnostics.Process GetProcessById(
   int processId
);
public static System.Diagnostics.Process GetProcessById(
   int processId,
   System.String machineName
);
public static System.Diagnostics.Process[] GetProcesses();
public static System.Diagnostics.Process[] GetProcesses(
   System.String machineName
);
public static System.Diagnostics.Process[] GetProcessesByName(
   System.String processName
);
public static System.Diagnostics.Process[] GetProcessesByName(
   System.String processName,
   System.String machineName
);
public void Kill();
public static void LeaveDebugMode();
public void Refresh();
public boolean Start();
public static System.Diagnostics.Process Start(
   System.Diagnostics.ProcessStartInfo startInfo
);
public static System.Diagnostics.Process Start(
   System.String fileName
);
public static System.Diagnostics.Process Start(
   System.String fileName,
   System.String arguments
);
public System.String ToString();
public void WaitForExit();
public boolean WaitForExit(
   int milliseconds
);
```

```
public boolean WaitForInputIdle();
public boolean WaitForInputIdle(
   int milliseconds
);
```

Public Events

```
public void add_Exited(
   System.EventHandler value
);
public void remove_Exited(
   System.EventHandler value
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void OnExited();
```

ProcessModule Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Diagnostics.Design.ProcessModuleDesigne
r, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class ProcessModule
   extends System.ComponentModel.Component
```

Public Properties

```
public System.IntPtr get_BaseAddress();
public System.IntPtr get_EntryPointAddress();
public System.String get_FileName();
public System.Diagnostics.FileVersionInfo get_FileVersionInfo();
public int get_ModuleMemorySize();
public System.String get_ModuleName();
```

Public Methods

```
public System.String ToString();
```

ProcessModuleCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ProcessModuleCollection
   extends System.Collections.ReadOnlyCollectionBase
```

Public Constructors

```
public ProcessModuleCollection(
   System.Diagnostics.ProcessModule[] processModules
);
```

Public Properties

```
public System.Diagnostics.ProcessModule get_Item(
   int index
);
```

Public Methods

```
public boolean Contains(
   System.Diagnostics.ProcessModule module
);
public void CopyTo(
   System.Diagnostics.ProcessModule[] array,
   int index
);
public int IndexOf(
   System.Diagnostics.ProcessModule module
);
```

Protected Constructors

```
protected ProcessModuleCollection();
```

ProcessPriorityClass Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ProcessPriorityClass
```

Members

AboveNormal

BelowNormal

High

Idle

Normal

RealTime

ProcessStartInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Compo
nentModel.ExpandableObjectConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyTo
ken=b77a5c561934e089") */
public class ProcessStartInfo
   extends System.Object
```

Public Constructors

```
public ProcessStartInfo();
public ProcessStartInfo(
   System.String fileName
);
public ProcessStartInfo(
   System.String fileName,
   System.String arguments
);
```

Public Properties

```
public System.String get_Arguments();
public void set_Arguments(
   System.String value
);
public boolean get_CreateNoWindow();
public void set_CreateNoWindow(
   boolean value
);
public System.Collections.Specialized.StringDictionary get_EnvironmentVariables();
public boolean get_ErrorDialog();
public void set_ErrorDialog(
   boolean value
);
public System.IntPtr get_ErrorDialogParentHandle();
public void set_ErrorDialogParentHandle(
   System.IntPtr value
);
public System.String get_FileName();
public void set_FileName(
   System.String value
);
public boolean get_RedirectStandardError();
public void set_RedirectStandardError(
   boolean value
);
public boolean get_RedirectStandardInput();
public void set_RedirectStandardInput(
   boolean value
);
public boolean get_RedirectStandardOutput();
public void set_RedirectStandardOutput(
   boolean value
);
public boolean get_UseShellExecute();
public void set_UseShellExecute(
   boolean value
```

```
);
public System.String get_Verb();
public void set_Verb(
   System.String value
);
public System.String[] get_Verbs();
public System.Diagnostics.ProcessWindowStyle get_WindowStyle();
public void set_WindowStyle(
   System.Diagnostics.ProcessWindowStyle value
);
public System.String get_WorkingDirectory();
public void set_WorkingDirectory(
   System.String value
);
```

ProcessThread Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Diagnostics.Design.ProcessThreadDesigne
r, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class ProcessThread
   extends System.ComponentModel.Component
```

Public Properties

```
public int get_BasePriority();
public int get_CurrentPriority();
public int get_Id();
public void set_IdealProcessor(
   int value
);
public boolean get_PriorityBoostEnabled();
public void set_PriorityBoostEnabled(
   boolean value
);
public System.Diagnostics.ThreadPriorityLevel get_PriorityLevel();
public void set_PriorityLevel(
   System.Diagnostics.ThreadPriorityLevel value
);
public System.TimeSpan get_PrivilegedProcessorTime();
public void set_ProcessorAffinity(
   System.IntPtr value
);
public System.IntPtr get_StartAddress();
public System.DateTime get_StartTime();
public System.Diagnostics.ThreadState get_ThreadState();
public System.TimeSpan get_TotalProcessorTime();
public System.TimeSpan get_UserProcessorTime();
public System.Diagnostics.ThreadWaitReason get_WaitReason();
```

Public Methods

```
public void ResetIdealProcessor();
```

ProcessThreadCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ProcessThreadCollection
   extends System.Collections.ReadOnlyCollectionBase
```

Public Constructors

```
public ProcessThreadCollection(
   System.Diagnostics.ProcessThread[] processThreads
);
```

Public Properties

```
public System.Diagnostics.ProcessThread get_Item(
   int index
);
```

Public Methods

```
public int Add(
   System.Diagnostics.ProcessThread thread
);
public boolean Contains(
   System.Diagnostics.ProcessThread thread
);
public void CopyTo(
   System.Diagnostics.ProcessThread[] array,
   int index
);
public int IndexOf(
   System.Diagnostics.ProcessThread thread
);
public void Insert(
   int index,
   System.Diagnostics.ProcessThread thread
);
public void Remove(
   System.Diagnostics.ProcessThread thread
);
```

Protected Constructors

```
protected ProcessThreadCollection();
```

ProcessWindowStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ProcessWindowStyle
```

Members

Hidden

Maximized

Minimized

Normal

StackFrame Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StackFrame
   extends System.Object
```

Public Constructors

```
public StackFrame();
public StackFrame(
   boolean fNeedFileInfo
);
public StackFrame(
   int skipFrames
);
public StackFrame(
   int skipFrames,
   boolean fNeedFileInfo
);
public StackFrame(
   System.String fileName,
   int lineNumber
);
public StackFrame(
   System.String fileName,
   int lineNumber,
   int colNumber
);
```

Public Methods

```
public int GetFileColumnNumber();
public int GetFileLineNumber();
public System.String GetFileName();
public int GetILOffset();
public System.Reflection.MethodBase GetMethod();
public int GetNativeOffset();
public System.String ToString();
```

Public Fields

```
public static int OFFSET_UNKNOWN;
```

StackTrace Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StackTrace
   extends System.Object
```

Public Constructors

```
public StackTrace();
public StackTrace(
   boolean fNeedFileInfo
);
public StackTrace(
   int skipFrames
);
public StackTrace(
   System.Diagnostics.StackFrame frame
);
public StackTrace(
   System.Exception e
);
public StackTrace(
   int skipFrames,
   boolean fNeedFileInfo
);
public StackTrace(
   System.Exception e,
   boolean fNeedFileInfo
);
public StackTrace(
   System.Exception e,
   int skipFrames
);
public StackTrace(
   System.Threading.Thread targetThread,
   boolean needFileInfo
);
public StackTrace(
   System.Exception e,
   int skipFrames,
   boolean fNeedFileInfo
);
```

Public Properties

```
public int get_FrameCount();
```

Public Methods

```
public System.Diagnostics.StackFrame GetFrame(
   int index
);
public System.String ToString();
```

Public Fields

```
public static int METHODS_TO_SKIP;
```

Switch Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class Switch
   extends System.Object
```

Public Properties

```
public System.String get_Description();
public System.String get_DisplayName();
```

Protected Constructors

```
protected Switch(
   System.String displayName,
   System.String description
);
```

Protected Properties

```
protected int get_SwitchSetting();
protected void set_SwitchSetting(
   int value
);
```

Protected Methods

```
protected void OnSwitchSettingChanged();
```

TextWriterTraceListener Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TextWriterTraceListener
   extends System.Diagnostics.TraceListener
```

Public Constructors

```
public TextWriterTraceListener();
public TextWriterTraceListener(
   System.IO.Stream stream
);
public TextWriterTraceListener(
   System.IO.TextWriter writer
);
public TextWriterTraceListener(
   System.String fileName
);
public TextWriterTraceListener(
   System.IO.Stream stream,
   System.String name
);
public TextWriterTraceListener(
   System.IO.TextWriter writer,
   System.String name
);
public TextWriterTraceListener(
   System.String fileName,
   System.String name
);
```

Public Properties

```
public System.IO.TextWriter get_Writer();
public void set_Writer(
   System.IO.TextWriter value
);
```

Public Methods

```
public void Close();
public void Flush();
public void Write(
   System.String message
);
public void WriteLine(
   System.String message
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

ThreadPriorityLevel Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ThreadPriorityLevel
```

Members

AboveNormal

BelowNormal

Highest

Idle

Lowest

Normal

TimeCritical

ThreadState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ThreadState
```

Members

Initialized

Ready

Running

Standby

Terminated

Transition

Unknown

Wait

ThreadWaitReason Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ThreadWaitReason
```

Members

EventPairHigh

EventPairLow

ExecutionDelay

Executive

FreePage

LpcReceive

LpcReply

PageIn

PageOut

Suspended

SystemAllocation

Unknown

UserRequest

VirtualMemory

Trace Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Trace
   extends System.Object
```

Public Properties

```
public static boolean get_AutoFlush();
public static void set_AutoFlush(
   boolean value
);
public static int get_IndentLevel();
public static void set_IndentLevel(
   int value
);
public static int get_IndentSize();
public static void set_IndentSize(
   int value
);
public static System.Diagnostics.TraceListenerCollection get_Listeners();
```

Public Methods

```
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void Assert(
   boolean condition
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void Assert(
   boolean condition,
   System.String message
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void Assert(
   boolean condition,
   System.String message,
   System.String detailMessage
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void Close();
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void Fail(
   System.String message
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void Fail(
   System.String message,
   System.String detailMessage
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void Flush();
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void Indent();
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void Unindent();
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void Write(
```

```
   System.Object value
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void Write(
   System.String message
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void Write(
   System.Object value,
   System.String category
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void Write(
   System.String message,
   System.String category
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void WriteIf(
   boolean condition,
   System.Object value
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void WriteIf(
   boolean condition,
   System.String message
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void WriteIf(
   boolean condition,
   System.Object value,
   System.String category
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void WriteIf(
   boolean condition,
   System.String message,
   System.String category
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void WriteLine(
   System.Object value
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void WriteLine(
   System.String message
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void WriteLine(
   System.Object value,
   System.String category
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void WriteLine(
   System.String message,
   System.String category
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void WriteLineIf(
   boolean condition,
   System.Object value
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void WriteLineIf(
   boolean condition,
   System.String message
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void WriteLineIf(
```

```
   boolean condition,
   System.Object value,
   System.String category
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "TRACE") */
public static void WriteLineIf(
   boolean condition,
   System.String message,
   System.String category
);
```

TraceLevel Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TraceLevel
```

Members

Error

Info

Off

Verbose

Warning

TraceListener Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class TraceListener
   extends System.MarshalByRefObject
   implements System.IDisposable
```

Public Properties

```
public int get_IndentLevel();
public void set_IndentLevel(
   int value
);
public int get_IndentSize();
public void set_IndentSize(
   int value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
```

Public Methods

```
public void Close();
public final void Dispose();
public void Fail(
   System.String message
);
public void Fail(
   System.String message,
   System.String detailMessage
);
public void Flush();
public void Write(
   System.Object o
);
public abstract void Write(
   System.String message
);
public void Write(
   System.Object o,
   System.String category
);
public void Write(
   System.String message,
   System.String category
);
public void WriteLine(
   System.Object o
);
public abstract void WriteLine(
   System.String message
);
public void WriteLine(
   System.Object o,
   System.String category
);
```

```
public void WriteLine(
   System.String message,
   System.String category
);
```

Protected Constructors

```
protected TraceListener();
protected TraceListener(
   System.String name
);
```

Protected Properties

```
protected boolean get_NeedIndent();
protected void set_NeedIndent(
   boolean value
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void WriteIndent();
```

TraceListenerCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class TraceListenerCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Properties

```
public final int get_Count();
public System.Diagnostics.TraceListener get_Item(
   int i
);
public System.Diagnostics.TraceListener get_Item(
   System.String name
);
public void set_Item(
   int i,
   System.Diagnostics.TraceListener value
);
```

Public Methods

```
public int Add(
   System.Diagnostics.TraceListener listener
);
public void AddRange(
   System.Diagnostics.TraceListenerCollection value
);
public void AddRange(
   System.Diagnostics.TraceListener[] value
);
public final void Clear();
public boolean Contains(
   System.Diagnostics.TraceListener listener
);
public void CopyTo(
   System.Diagnostics.TraceListener[] listeners,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Diagnostics.TraceListener listener
);
public void Insert(
   int index,
   System.Diagnostics.TraceListener listener
);
public void Remove(
   System.Diagnostics.TraceListener listener
);
public void Remove(
   System.String name
);
public final void RemoveAt(
   int index
```

```
);
```

TraceSwitch Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TraceSwitch
   extends System.Diagnostics.Switch
```

Public Constructors

```
public TraceSwitch(
   System.String displayName,
   System.String description
);
```

Public Properties

```
public System.Diagnostics.TraceLevel get_Level();
public void set_Level(
   System.Diagnostics.TraceLevel value
);
public boolean get_TraceError();
public boolean get_TraceInfo();
public boolean get_TraceVerbose();
public boolean get_TraceWarning();
```

Protected Methods

```
protected void OnSwitchSettingChanged();
```

System.Diagnostics.SymbolStore Namespace (Visual J# Syntax)

Classes

SymDocumentType

SymLanguageType

SymLanguageVendor

Interfaces

ISymbolBinder

ISymbolDocument

ISymbolDocumentWriter

ISymbolMethod

ISymbolNamespace

ISymbolReader

ISymbolScope

ISymbolVariable

ISymbolWriter

Structures

SymbolToken

Enumerations

SymAddressKind

ISymbolBinder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISymbolBinder
```

Public Methods

```
System.Diagnostics.SymbolStore.ISymbolReader GetReader(
   int importer,
   System.String filename,
   System.String searchPath
);
```

ISymbolDocument Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISymbolDocument
```

Public Properties

```
System.Guid get_CheckSumAlgorithmId();
System.Guid get_DocumentType();
boolean get_HasEmbeddedSource();
System.Guid get_Language();
System.Guid get_LanguageVendor();
int get_SourceLength();
System.String get_URL();
```

Public Methods

```
int FindClosestLine(
   int line
);
ubyte[] GetCheckSum();
ubyte[] GetSourceRange(
   int startLine,
   int startColumn,
   int endLine,
   int endColumn
);
```

ISymbolDocumentWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISymbolDocumentWriter
```

Public Methods

```
void SetCheckSum(
   System.Guid algorithmId,
   ubyte[] checkSum
);
void SetSource(
   ubyte[] source
);
```

ISymbolMethod Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISymbolMethod
```

Public Properties

```
System.Diagnostics.SymbolStore.ISymbolScope get_RootScope();
int get_SequencePointCount();
System.Diagnostics.SymbolStore.SymbolToken get_Token();
```

Public Methods

```
System.Diagnostics.SymbolStore.ISymbolNamespace GetNamespace();
int GetOffset(
   System.Diagnostics.SymbolStore.ISymbolDocument document,
   int line,
   int column
);
System.Diagnostics.SymbolStore.ISymbolVariable[] GetParameters();
int[] GetRanges(
   System.Diagnostics.SymbolStore.ISymbolDocument document,
   int line,
   int column
);
System.Diagnostics.SymbolStore.ISymbolScope GetScope(
   int offset
);
void GetSequencePoints(
   int[] offsets,
   System.Diagnostics.SymbolStore.ISymbolDocument[] documents,
   int[] lines,
   int[] columns,
   int[] endLines,
   int[] endColumns
);
boolean GetSourceStartEnd(
   System.Diagnostics.SymbolStore.ISymbolDocument[] docs,
   int[] lines,
   int[] columns
);
```

ISymbolNamespace Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISymbolNamespace
```

Public Properties

```
System.String get_Name();
```

Public Methods

```
System.Diagnostics.SymbolStore.ISymbolNamespace[] GetNamespaces();
System.Diagnostics.SymbolStore.ISymbolVariable[] GetVariables();
```

ISymbolReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISymbolReader
```

Public Properties

```
System.Diagnostics.SymbolStore.SymbolToken get_UserEntryPoint();
```

Public Methods

```
System.Diagnostics.SymbolStore.ISymbolDocument GetDocument(
   System.String url,
   System.Guid language,
   System.Guid languageVendor,
   System.Guid documentType
);
System.Diagnostics.SymbolStore.ISymbolDocument[] GetDocuments();
System.Diagnostics.SymbolStore.ISymbolVariable[] GetGlobalVariables();
System.Diagnostics.SymbolStore.ISymbolMethod GetMethod(
   System.Diagnostics.SymbolStore.SymbolToken method
);
System.Diagnostics.SymbolStore.ISymbolMethod GetMethod(
   System.Diagnostics.SymbolStore.SymbolToken method,
   int version
);
System.Diagnostics.SymbolStore.ISymbolMethod GetMethodFromDocumentPosition(
   System.Diagnostics.SymbolStore.ISymbolDocument document,
   int line,
   int column
);
System.Diagnostics.SymbolStore.ISymbolNamespace[] GetNamespaces();
ubyte[] GetSymAttribute(
   System.Diagnostics.SymbolStore.SymbolToken parent,
   System.String name
);
System.Diagnostics.SymbolStore.ISymbolVariable[] GetVariables(
   System.Diagnostics.SymbolStore.SymbolToken parent
);
```

ISymbolScope Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISymbolScope
```

Public Properties

```
int get_EndOffset();
System.Diagnostics.SymbolStore.ISymbolMethod get_Method();
System.Diagnostics.SymbolStore.ISymbolScope get_Parent();
int get_StartOffset();
```

Public Methods

```
System.Diagnostics.SymbolStore.ISymbolScope[] GetChildren();
System.Diagnostics.SymbolStore.ISymbolVariable[] GetLocals();
System.Diagnostics.SymbolStore.ISymbolNamespace[] GetNamespaces();
```

ISymbolVariable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISymbolVariable
```

Public Properties

```
int get_AddressField1();
int get_AddressField2();
int get_AddressField3();
System.Diagnostics.SymbolStore.SymAddressKind get_AddressKind();
System.Object get_Attributes();
int get_EndOffset();
System.String get_Name();
int get_StartOffset();
```

Public Methods

```
ubyte[] GetSignature();
```

ISymbolWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISymbolWriter
```

Public Methods

```
void Close();
void CloseMethod();
void CloseNamespace();
void CloseScope(
   int endOffset
);
System.Diagnostics.SymbolStore.ISymbolDocumentWriter DefineDocument(
   System.String url,
   System.Guid language,
   System.Guid languageVendor,
   System.Guid documentType
);
void DefineField(
   System.Diagnostics.SymbolStore.SymbolToken parent,
   System.String name,
   System.Reflection.FieldAttributes attributes,
   ubyte[] signature,
   System.Diagnostics.SymbolStore.SymAddressKind addrKind,
   int addr1,
   int addr2,
   int addr3
);
void DefineGlobalVariable(
   System.String name,
   System.Reflection.FieldAttributes attributes,
   ubyte[] signature,
   System.Diagnostics.SymbolStore.SymAddressKind addrKind,
   int addr1,
   int addr2,
   int addr3
);
void DefineLocalVariable(
   System.String name,
   System.Reflection.FieldAttributes attributes,
   ubyte[] signature,
   System.Diagnostics.SymbolStore.SymAddressKind addrKind,
   int addr1,
   int addr2,
   int addr3,
   int startOffset,
   int endOffset
);
void DefineParameter(
   System.String name,
   System.Reflection.ParameterAttributes attributes,
   int sequence,
   System.Diagnostics.SymbolStore.SymAddressKind addrKind,
   int addr1,
   int addr2,
   int addr3
);
void DefineSequencePoints(
   System.Diagnostics.SymbolStore.ISymbolDocumentWriter document,
```

```
      int[] offsets,
      int[] lines,
      int[] columns,
      int[] endLines,
      int[] endColumns
   );
   void Initialize(
      System.IntPtr emitter,
      System.String filename,
      boolean fFullBuild
   );
   void OpenMethod(
      System.Diagnostics.SymbolStore.SymbolToken method
   );
   void OpenNamespace(
      System.String name
   );
   int OpenScope(
      int startOffset
   );
   void SetMethodSourceRange(
      System.Diagnostics.SymbolStore.ISymbolDocumentWriter startDoc,
      int startLine,
      int startColumn,
      System.Diagnostics.SymbolStore.ISymbolDocumentWriter endDoc,
      int endLine,
      int endColumn
   );
   void SetScopeRange(
      int scopeID,
      int startOffset,
      int endOffset
   );
   void SetSymAttribute(
      System.Diagnostics.SymbolStore.SymbolToken parent,
      System.String name,
      ubyte[] data
   );
   void SetUnderlyingWriter(
      System.IntPtr underlyingWriter
   );
   void SetUserEntryPoint(
      System.Diagnostics.SymbolStore.SymbolToken entryMethod
   );
   void UsingNamespace(
      System.String fullName
   );
```

SymAddressKind Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SymAddressKind
```

Members

BitField

ILOffset

NativeOffset

NativeRegister

NativeRegisterRegister

NativeRegisterRelative

NativeRegisterStack

NativeRVA

NativeStackRegister

SymbolToken Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SymbolToken
   extends System.ValueType
```

Public Constructors

```
public SymbolToken(
   int val
);
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public int GetToken();
```

SymDocumentType Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SymDocumentType
   extends System.Object
```

Public Constructors

```
public SymDocumentType();
```

Public Fields

```
public static final System.Guid Text;
```

SymLanguageType Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SymLanguageType
   extends System.Object
```

Public Constructors

```
public SymLanguageType();
```

Public Fields

```
public static final System.Guid Basic;
public static final System.Guid C;
public static final System.Guid Cobol;
public static final System.Guid CPlusPlus;
public static final System.Guid CSharp;
public static final System.Guid ILAssembly;
public static final System.Guid Java;
public static final System.Guid JScript;
public static final System.Guid MCPlusPlus;
public static final System.Guid Pascal;
public static final System.Guid SMC;
```

SymLanguageVendor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SymLanguageVendor
   extends System.Object
```

Public Constructors

```
public SymLanguageVendor();
```

Public Fields

```
public static final System.Guid Microsoft;
```

System.DirectoryServices Namespace (Visual J# Syntax)

Classes

DirectoryEntries

DirectoryEntry

DirectorySearcher

DirectoryServicesPermission

DirectoryServicesPermissionAttribute

DirectoryServicesPermissionEntry

DirectoryServicesPermissionEntryCollection

DSDescriptionAttribute

PropertyCollection

PropertyValueCollection

ResultPropertyCollection

ResultPropertyValueCollection

SchemaNameCollection

SearchResult

SearchResultCollection

SortOption

Enumerations

AuthenticationTypes

DirectoryServicesPermissionAccess

ReferralChasingOption

SearchScope

SortDirection

AuthenticationTypes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum AuthenticationTypes
```

Members

Anonymous

Delegation

Encryption

FastBind

None

ReadonlyServer

Sealing

Secure

SecureSocketsLayer

ServerBind

Signing

DirectoryEntries Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DirectoryEntries
   extends System.Object
   implements System.Collections.IEnumerable
```

Public Properties

```
public System.DirectoryServices.SchemaNameCollection get_SchemaFilter();
```

Public Methods

```
public System.DirectoryServices.DirectoryEntry Add(
   System.String name,
   System.String schemaClassName
);
public System.DirectoryServices.DirectoryEntry Find(
   System.String name
);
public System.DirectoryServices.DirectoryEntry Find(
   System.String name,
   System.String schemaClassName
);
public final System.Collections.IEnumerator GetEnumerator();
public void Remove(
   System.DirectoryServices.DirectoryEntry entry
);
```

DirectoryEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Direc
toryServices.Design.DirectoryEntryConverter, System.DirectoryServices, Version=1.0.3300.0, Cu
lture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class DirectoryEntry
   extends System.ComponentModel.Component
```

Public Constructors

```
public DirectoryEntry();
public DirectoryEntry(
   System.Object adsObject
);
public DirectoryEntry(
   System.String path
);
public DirectoryEntry(
   System.String path,
   System.String username,
   System.String password
);
public DirectoryEntry(
   System.String path,
   System.String username,
   System.String password,
   System.DirectoryServices.AuthenticationTypes authenticationType
);
```

Public Properties

```
public System.DirectoryServices.AuthenticationTypes get_AuthenticationType();
public void set_AuthenticationType(
   System.DirectoryServices.AuthenticationTypes value
);
public System.DirectoryServices.DirectoryEntries get_Children();
public System.Guid get_Guid();
public System.String get_Name();
public System.String get_NativeGuid();
public System.Object get_NativeObject();
public System.DirectoryServices.DirectoryEntry get_Parent();
public System.String get_Password();
public void set_Password(
   System.String value
);
public System.String get_Path();
public void set_Path(
   System.String value
);
public System.DirectoryServices.PropertyCollection get_Properties();
public System.String get_SchemaClassName();
public System.DirectoryServices.DirectoryEntry get_SchemaEntry();
public boolean get_UsePropertyCache();
public void set_UsePropertyCache(
   boolean value
);
public System.String get_Username();
```

```
public void set_Username(
   System.String value
);
```

Public Methods

```
public void Close();
public void CommitChanges();
public System.DirectoryServices.DirectoryEntry CopyTo(
   System.DirectoryServices.DirectoryEntry newParent
);
public System.DirectoryServices.DirectoryEntry CopyTo(
   System.DirectoryServices.DirectoryEntry newParent,
   System.String newName
);
public void DeleteTree();
public static boolean Exists(
   System.String path
);
public System.Object Invoke(
   System.String methodName,
   System.Object[] args
);
public void MoveTo(
   System.DirectoryServices.DirectoryEntry newParent
);
public void MoveTo(
   System.DirectoryServices.DirectoryEntry newParent,
   System.String newName
);
public void RefreshCache();
public void RefreshCache(
   System.String[] propertyNames
);
public void Rename(
   System.String newName
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

DirectorySearcher Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DirectorySearcher
   extends System.ComponentModel.Component
```

Public Constructors

```
public DirectorySearcher();
public DirectorySearcher(
   System.DirectoryServices.DirectoryEntry searchRoot
);
public DirectorySearcher(
   System.String filter
);
public DirectorySearcher(
   System.DirectoryServices.DirectoryEntry searchRoot,
   System.String filter
);
public DirectorySearcher(
   System.String filter,
   System.String[] propertiesToLoad
);
public DirectorySearcher(
   System.DirectoryServices.DirectoryEntry searchRoot,
   System.String filter,
   System.String[] propertiesToLoad
);
public DirectorySearcher(
   System.String filter,
   System.String[] propertiesToLoad,
   System.DirectoryServices.SearchScope scope
);
public DirectorySearcher(
   System.DirectoryServices.DirectoryEntry searchRoot,
   System.String filter,
   System.String[] propertiesToLoad,
   System.DirectoryServices.SearchScope scope
);
```

Public Properties

```
public boolean get_CacheResults();
public void set_CacheResults(
   boolean value
);
public System.TimeSpan get_ClientTimeout();
public void set_ClientTimeout(
   System.TimeSpan value
);
public System.String get_Filter();
public void set_Filter(
   System.String value
);
public int get_PageSize();
public void set_PageSize(
   int value
);
```

```
public System.Collections.Specialized.StringCollection get_PropertiesToLoad();
public boolean get_PropertyNamesOnly();
public void set_PropertyNamesOnly(
   boolean value
);
public System.DirectoryServices.ReferralChasingOption get_ReferralChasing();
public void set_ReferralChasing(
   System.DirectoryServices.ReferralChasingOption value
);
public System.DirectoryServices.DirectoryEntry get_SearchRoot();
public void set_SearchRoot(
   System.DirectoryServices.DirectoryEntry value
);
public System.DirectoryServices.SearchScope get_SearchScope();
public void set_SearchScope(
   System.DirectoryServices.SearchScope value
);
public System.TimeSpan get_ServerPageTimeLimit();
public void set_ServerPageTimeLimit(
   System.TimeSpan value
);
public System.TimeSpan get_ServerTimeLimit();
public void set_ServerTimeLimit(
   System.TimeSpan value
);
public int get_SizeLimit();
public void set_SizeLimit(
   int value
);
public System.DirectoryServices.SortOption get_Sort();
public void set_Sort(
   System.DirectoryServices.SortOption value
);
```

Public Methods

```
public System.DirectoryServices.SearchResultCollection FindAll();
public System.DirectoryServices.SearchResult FindOne();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

DirectoryServicesPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DirectoryServicesPermission
   extends System.Security.Permissions.ResourcePermissionBase
```

Public Constructors

```
public DirectoryServicesPermission();
public DirectoryServicesPermission(
   System.DirectoryServices.DirectoryServicesPermissionEntry[] permissionAccessEntries
);
public DirectoryServicesPermission(
   System.Security.Permissions.PermissionState state
);
public DirectoryServicesPermission(
   System.DirectoryServices.DirectoryServicesPermissionAccess permissionAccess,
   System.String path
);
```

Public Properties

```
public System.DirectoryServices.DirectoryServicesPermissionEntryCollection get_PermissionEntr
ies();
```

DirectoryServicesPermissionAccess Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum DirectoryServicesPermissionAccess
```

Members

Browse

None

Write

DirectoryServicesPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, Event, AllowMultiple = true, Inherited = false) */
public class DirectoryServicesPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public DirectoryServicesPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.String get_Path();
public void set_Path(
   System.String value
);
public System.DirectoryServices.DirectoryServicesPermissionAccess get_PermissionAccess();
public void set_PermissionAccess(
   System.DirectoryServices.DirectoryServicesPermissionAccess value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

DirectoryServicesPermissionEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DirectoryServicesPermissionEntry
   extends System.Object
```

Public Constructors

```
public DirectoryServicesPermissionEntry(
   System.DirectoryServices.DirectoryServicesPermissionAccess permissionAccess,
   System.String path
);
```

Public Properties

```
public System.String get_Path();
public System.DirectoryServices.DirectoryServicesPermissionAccess get_PermissionAccess();
```

DirectoryServicesPermissionEntryCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DirectoryServicesPermissionEntryCollection
   extends System.Collections.CollectionBase
```

Public Properties

```
public System.DirectoryServices.DirectoryServicesPermissionEntry get_Item(
   int index
);
public void set_Item(
   int index,
   System.DirectoryServices.DirectoryServicesPermissionEntry value
);
```

Public Methods

```
public int Add(
   System.DirectoryServices.DirectoryServicesPermissionEntry value
);
public void AddRange(
   System.DirectoryServices.DirectoryServicesPermissionEntryCollection value
);
public void AddRange(
   System.DirectoryServices.DirectoryServicesPermissionEntry[] value
);
public boolean Contains(
   System.DirectoryServices.DirectoryServicesPermissionEntry value
);
public void CopyTo(
   System.DirectoryServices.DirectoryServicesPermissionEntry[] array,
   int index
);
public int IndexOf(
   System.DirectoryServices.DirectoryServicesPermissionEntry value
);
public void Insert(
   int index,
   System.DirectoryServices.DirectoryServicesPermissionEntry value
);
public void Remove(
   System.DirectoryServices.DirectoryServicesPermissionEntry value
);
```

Protected Methods

```
protected void OnClear();
protected void OnInsert(
   int index,
   System.Object value
```

```
);
protected void OnRemove(
   int index,
   System.Object value
);
protected void OnSet(
   int index,
   System.Object oldValue,
   System.Object newValue
);
```

DSDescriptionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class DSDescriptionAttribute
   extends System.ComponentModel.DescriptionAttribute
```

Public Constructors

```
public DSDescriptionAttribute(
   System.String description
);
```

Public Properties

```
public System.String get_Description();
```

PropertyCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class PropertyCollection
   extends System.Object
   implements System.Collections.IDictionary, System.Collections.ICollection, System.Collecti
ons.IEnumerable
```

Public Properties

```
public final int get_Count();
public System.DirectoryServices.PropertyValueCollection get_Item(
   System.String propertyName
);
public System.Collections.ICollection get_PropertyNames();
public final System.Collections.ICollection get_Values();
```

Public Methods

```
public boolean Contains(
   System.String propertyName
);
public void CopyTo(
   System.DirectoryServices.PropertyValueCollection[] array,
   int index
);
public final System.Collections.IDictionaryEnumerator GetEnumerator();
```

PropertyValueCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class PropertyValueCollection
   extends System.Collections.CollectionBase
```

Public Properties

```
public System.Object get_Item(
   int index
);
public void set_Item(
   int index,
   System.Object value
);
public System.Object get_Value();
public void set_Value(
   System.Object value
);
```

Public Methods

```
public int Add(
   System.Object value
);
public void AddRange(
   System.DirectoryServices.PropertyValueCollection value
);
public void AddRange(
   System.Object[] value
);
public boolean Contains(
   System.Object value
);
public void CopyTo(
   System.Object[] array,
   int index
);
public int IndexOf(
   System.Object value
);
public void Insert(
   int index,
   System.Object value
);
public void Remove(
   System.Object value
);
```

Protected Methods

```
protected void OnClearComplete();
protected void OnInsertComplete(
   int index,
   System.Object value
```

```
);
protected void OnRemoveComplete(
   int index,
   System.Object value
);
protected void OnSetComplete(
   int index,
   System.Object oldValue,
   System.Object newValue
);
```

ReferralChasingOption Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ReferralChasingOption
```

Members

All

External

None

Subordinate

ResultPropertyCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ResultPropertyCollection
   extends System.Collections.DictionaryBase
```

Public Properties

```
public System.DirectoryServices.ResultPropertyValueCollection get_Item(
   System.String name
);
public System.Collections.ICollection get_PropertyNames();
public System.Collections.ICollection get_Values();
```

Public Methods

```
public boolean Contains(
   System.String propertyName
);
public void CopyTo(
   System.DirectoryServices.ResultPropertyValueCollection[] array,
   int index
);
```

ResultPropertyValueCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ResultPropertyValueCollection
   extends System.Collections.ReadOnlyCollectionBase
```

Public Properties

```
public System.Object get_Item(
   int index
);
```

Public Methods

```
public boolean Contains(
   System.Object value
);
public void CopyTo(
   System.Object[] values,
   int index
);
public int IndexOf(
   System.Object value
);
```

SchemaNameCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SchemaNameCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Properties

```
public final int get_Count();
public System.String get_Item(
   int index
);
public void set_Item(
   int index,
   System.String value
);
```

Public Methods

```
public int Add(
   System.String value
);
public void AddRange(
   System.DirectoryServices.SchemaNameCollection value
);
public void AddRange(
   System.String[] value
);
public final void Clear();
public boolean Contains(
   System.String value
);
public void CopyTo(
   System.String[] stringArray,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.String value
);
public void Insert(
   int index,
   System.String value
);
public void Remove(
   System.String value
);
public final void RemoveAt(
   int index
);
```

SearchResult Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SearchResult
   extends System.Object
```

Public Properties

```
public System.String get_Path();
public System.DirectoryServices.ResultPropertyCollection get_Properties();
```

Public Methods

```
public System.DirectoryServices.DirectoryEntry GetDirectoryEntry();
```

SearchResultCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SearchResultCollection
   extends System.MarshalByRefObject
   implements System.Collections.ICollection, System.Collections.IEnumerable, System.IDisposa
ble
```

Public Properties

```
public final int get_Count();
public System.IntPtr get_Handle();
public System.DirectoryServices.SearchResult get_Item(
   int index
);
public System.String[] get_PropertiesLoaded();
```

Public Methods

```
public boolean Contains(
   System.DirectoryServices.SearchResult result
);
public void CopyTo(
   System.DirectoryServices.SearchResult[] results,
   int index
);
public final void Dispose();
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.DirectoryServices.SearchResult result
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

SearchScope Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SearchScope
```

Members

Base

OneLevel

Subtree

SortDirection Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SortDirection
```

Members

Ascending

Descending

SortOption Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Compo
nentModel.ExpandableObjectConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyTo
ken=b77a5c561934e089") */
public class SortOption
   extends System.Object
```

Public Constructors

```
public SortOption();
public SortOption(
   System.String propertyName,
   System.DirectoryServices.SortDirection direction
);
```

Public Properties

```
public System.DirectoryServices.SortDirection get_Direction();
public void set_Direction(
   System.DirectoryServices.SortDirection value
);
public System.String get_PropertyName();
public void set_PropertyName(
   System.String value
);
```

System.Drawing Namespace (Visual J# Syntax)

Classes

Bitmap

Brush

Brushes

ColorConverter

ColorTranslator

Font

FontConverter

FontConverter.FontNameConverter

FontConverter.FontUnitConverter

FontFamily

Graphics

Icon

IconConverter

Image

ImageAnimator

ImageConverter

ImageFormatConverter

Pen

Pens

PointConverter

RectangleConverter

Region

SizeConverter

SolidBrush

StringFormat

SystemBrushes

SystemColors

SystemIcons

SystemPens

TextureBrush

ToolboxBitmapAttribute

Structures

CharacterRange

Color

Point

PointF

Rectangle

RectangleF

Size

SizeF

Delegates

Graphics.DrawImageAbort

Graphics.EnumerateMetafileProc

Image.GetThumbnailImageAbort

Enumerations

ContentAlignment

FontStyle

GraphicsUnit

KnownColor

RotateFlipType

StringAlignment

StringDigitSubstitute

StringFormatFlags

StringTrimming

StringUnit

Bitmap Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Drawing.Design.BitmapEditor, System.Drawing.Design, Vers
ion=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class Bitmap
   extends System.Drawing.Image
```

Public Constructors

```
public Bitmap(
   System.Drawing.Image original
);
public Bitmap(
   System.IO.Stream stream
);
public Bitmap(
   System.String filename
);
public Bitmap(
   int width,
   int height
);
public Bitmap(
   System.Drawing.Image original,
   System.Drawing.Size newSize
);
public Bitmap(
   System.IO.Stream stream,
   boolean useIcm
);
public Bitmap(
   System.String filename,
   boolean useIcm
);
public Bitmap(
   System.Type type,
   System.String resource
);
public Bitmap(
   int width,
   int height,
   System.Drawing.Graphics g
);
public Bitmap(
   int width,
   int height,
   System.Drawing.Imaging.PixelFormat format
);
public Bitmap(
   System.Drawing.Image original,
   int width,
   int height
);
public Bitmap(
   int width,
```

```
   int height,
   int stride,
   System.Drawing.Imaging.PixelFormat format,
   System.IntPtr scan0
);
```

Public Methods

```
public System.Drawing.Bitmap Clone(
   System.Drawing.Rectangle rect,
   System.Drawing.Imaging.PixelFormat format
);
public System.Drawing.Bitmap Clone(
   System.Drawing.RectangleF rect,
   System.Drawing.Imaging.PixelFormat format
);
public static System.Drawing.Bitmap FromHicon(
   System.IntPtr hicon
);
public static System.Drawing.Bitmap FromResource(
   System.IntPtr hinstance,
   System.String bitmapName
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public System.IntPtr GetHbitmap();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public System.IntPtr GetHbitmap(
   System.Drawing.Color background
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public System.IntPtr GetHicon();
public System.Drawing.Color GetPixel(
   int x,
   int y
);
public System.Drawing.Imaging.BitmapData LockBits(
   System.Drawing.Rectangle rect,
   System.Drawing.Imaging.ImageLockMode flags,
   System.Drawing.Imaging.PixelFormat format
);
public void MakeTransparent();
public void MakeTransparent(
   System.Drawing.Color transparentColor
);
public void SetPixel(
   int x,
   int y,
   System.Drawing.Color color
);
public void SetResolution(
   float xDpi,
   float yDpi
);
public void UnlockBits(
   System.Drawing.Imaging.BitmapData bitmapdata
);
```

Brush Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class Brush
   extends System.MarshalByRefObject
   implements System.ICloneable, System.IDisposable
```

Public Methods

```
public abstract System.Object Clone();
public final void Dispose();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

Brushes Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Brushes
   extends System.Object
```

Public Properties

```
public static System.Drawing.Brush get_AliceBlue();
public static System.Drawing.Brush get_AntiqueWhite();
public static System.Drawing.Brush get_Aqua();
public static System.Drawing.Brush get_Aquamarine();
public static System.Drawing.Brush get_Azure();
public static System.Drawing.Brush get_Beige();
public static System.Drawing.Brush get_Bisque();
public static System.Drawing.Brush get_Black();
public static System.Drawing.Brush get_BlanchedAlmond();
public static System.Drawing.Brush get_Blue();
public static System.Drawing.Brush get_BlueViolet();
public static System.Drawing.Brush get_Brown();
public static System.Drawing.Brush get_BurlyWood();
public static System.Drawing.Brush get_CadetBlue();
public static System.Drawing.Brush get_Chartreuse();
public static System.Drawing.Brush get_Chocolate();
public static System.Drawing.Brush get_Coral();
public static System.Drawing.Brush get_CornflowerBlue();
public static System.Drawing.Brush get_Cornsilk();
public static System.Drawing.Brush get_Crimson();
public static System.Drawing.Brush get_Cyan();
public static System.Drawing.Brush get_DarkBlue();
public static System.Drawing.Brush get_DarkCyan();
public static System.Drawing.Brush get_DarkGoldenrod();
public static System.Drawing.Brush get_DarkGray();
public static System.Drawing.Brush get_DarkGreen();
public static System.Drawing.Brush get_DarkKhaki();
public static System.Drawing.Brush get_DarkMagenta();
public static System.Drawing.Brush get_DarkOliveGreen();
public static System.Drawing.Brush get_DarkOrange();
public static System.Drawing.Brush get_DarkOrchid();
public static System.Drawing.Brush get_DarkRed();
public static System.Drawing.Brush get_DarkSalmon();
public static System.Drawing.Brush get_DarkSeaGreen();
public static System.Drawing.Brush get_DarkSlateBlue();
public static System.Drawing.Brush get_DarkSlateGray();
public static System.Drawing.Brush get_DarkTurquoise();
public static System.Drawing.Brush get_DarkViolet();
public static System.Drawing.Brush get_DeepPink();
public static System.Drawing.Brush get_DeepSkyBlue();
public static System.Drawing.Brush get_DimGray();
public static System.Drawing.Brush get_DodgerBlue();
public static System.Drawing.Brush get_Firebrick();
public static System.Drawing.Brush get_FloralWhite();
public static System.Drawing.Brush get_ForestGreen();
public static System.Drawing.Brush get_Fuchsia();
public static System.Drawing.Brush get_Gainsboro();
public static System.Drawing.Brush get_GhostWhite();
public static System.Drawing.Brush get_Gold();
public static System.Drawing.Brush get_Goldenrod();
public static System.Drawing.Brush get_Gray();
public static System.Drawing.Brush get_Green();
```

```
public static System.Drawing.Brush get_GreenYellow();
public static System.Drawing.Brush get_Honeydew();
public static System.Drawing.Brush get_HotPink();
public static System.Drawing.Brush get_IndianRed();
public static System.Drawing.Brush get_Indigo();
public static System.Drawing.Brush get_Ivory();
public static System.Drawing.Brush get_Khaki();
public static System.Drawing.Brush get_Lavender();
public static System.Drawing.Brush get_LavenderBlush();
public static System.Drawing.Brush get_LawnGreen();
public static System.Drawing.Brush get_LemonChiffon();
public static System.Drawing.Brush get_LightBlue();
public static System.Drawing.Brush get_LightCoral();
public static System.Drawing.Brush get_LightCyan();
public static System.Drawing.Brush get_LightGoldenrodYellow();
public static System.Drawing.Brush get_LightGray();
public static System.Drawing.Brush get_LightGreen();
public static System.Drawing.Brush get_LightPink();
public static System.Drawing.Brush get_LightSalmon();
public static System.Drawing.Brush get_LightSeaGreen();
public static System.Drawing.Brush get_LightSkyBlue();
public static System.Drawing.Brush get_LightSlateGray();
public static System.Drawing.Brush get_LightSteelBlue();
public static System.Drawing.Brush get_LightYellow();
public static System.Drawing.Brush get_Lime();
public static System.Drawing.Brush get_LimeGreen();
public static System.Drawing.Brush get_Linen();
public static System.Drawing.Brush get_Magenta();
public static System.Drawing.Brush get_Maroon();
public static System.Drawing.Brush get_MediumAquamarine();
public static System.Drawing.Brush get_MediumBlue();
public static System.Drawing.Brush get_MediumOrchid();
public static System.Drawing.Brush get_MediumPurple();
public static System.Drawing.Brush get_MediumSeaGreen();
public static System.Drawing.Brush get_MediumSlateBlue();
public static System.Drawing.Brush get_MediumSpringGreen();
public static System.Drawing.Brush get_MediumTurquoise();
public static System.Drawing.Brush get_MediumVioletRed();
public static System.Drawing.Brush get_MidnightBlue();
public static System.Drawing.Brush get_MintCream();
public static System.Drawing.Brush get_MistyRose();
public static System.Drawing.Brush get_Moccasin();
public static System.Drawing.Brush get_NavajoWhite();
public static System.Drawing.Brush get_Navy();
public static System.Drawing.Brush get_OldLace();
public static System.Drawing.Brush get_Olive();
public static System.Drawing.Brush get_OliveDrab();
public static System.Drawing.Brush get_Orange();
public static System.Drawing.Brush get_OrangeRed();
public static System.Drawing.Brush get_Orchid();
public static System.Drawing.Brush get_PaleGoldenrod();
public static System.Drawing.Brush get_PaleGreen();
public static System.Drawing.Brush get_PaleTurquoise();
public static System.Drawing.Brush get_PaleVioletRed();
public static System.Drawing.Brush get_PapayaWhip();
public static System.Drawing.Brush get_PeachPuff();
public static System.Drawing.Brush get_Peru();
public static System.Drawing.Brush get_Pink();
public static System.Drawing.Brush get_Plum();
public static System.Drawing.Brush get_PowderBlue();
public static System.Drawing.Brush get_Purple();
public static System.Drawing.Brush get_Red();
public static System.Drawing.Brush get_RosyBrown();
public static System.Drawing.Brush get_RoyalBlue();
public static System.Drawing.Brush get_SaddleBrown();
public static System.Drawing.Brush get_Salmon();
public static System.Drawing.Brush get_SandyBrown();
public static System.Drawing.Brush get_SeaGreen();
```

```
public static System.Drawing.Brush get_SeaShell();
public static System.Drawing.Brush get_Sienna();
public static System.Drawing.Brush get_Silver();
public static System.Drawing.Brush get_SkyBlue();
public static System.Drawing.Brush get_SlateBlue();
public static System.Drawing.Brush get_SlateGray();
public static System.Drawing.Brush get_Snow();
public static System.Drawing.Brush get_SpringGreen();
public static System.Drawing.Brush get_SteelBlue();
public static System.Drawing.Brush get_Tan();
public static System.Drawing.Brush get_Teal();
public static System.Drawing.Brush get_Thistle();
public static System.Drawing.Brush get_Tomato();
public static System.Drawing.Brush get_Transparent();
public static System.Drawing.Brush get_Turquoise();
public static System.Drawing.Brush get_Violet();
public static System.Drawing.Brush get_Wheat();
public static System.Drawing.Brush get_White();
public static System.Drawing.Brush get_WhiteSmoke();
public static System.Drawing.Brush get_Yellow();
public static System.Drawing.Brush get_YellowGreen();
```

CharacterRange Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CharacterRange
   extends System.ValueType
```

Public Constructors

```
public CharacterRange(
   int First,
   int Length
);
```

Public Properties

```
public int get_First();
public void set_First(
   int value
);
public int get_Length();
public void set_Length(
   int value
);
```

Color Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Drawi
ng.ColorConverter, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f
7f11d50a3a") */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Drawing.Design.ColorEditor, System.Drawing.Design, Versi
on=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class Color
   extends System.ValueType
```

Public Properties

```
public ubyte get_A();
public static System.Drawing.Color get_AliceBlue();
public static System.Drawing.Color get_AntiqueWhite();
public static System.Drawing.Color get_Aqua();
public static System.Drawing.Color get_Aquamarine();
public static System.Drawing.Color get_Azure();
public ubyte get_B();
public static System.Drawing.Color get_Beige();
public static System.Drawing.Color get_Bisque();
public static System.Drawing.Color get_Black();
public static System.Drawing.Color get_BlanchedAlmond();
public static System.Drawing.Color get_Blue();
public static System.Drawing.Color get_BlueViolet();
public static System.Drawing.Color get_Brown();
public static System.Drawing.Color get_BurlyWood();
public static System.Drawing.Color get_CadetBlue();
public static System.Drawing.Color get_Chartreuse();
public static System.Drawing.Color get_Chocolate();
public static System.Drawing.Color get_Coral();
public static System.Drawing.Color get_CornflowerBlue();
public static System.Drawing.Color get_Cornsilk();
public static System.Drawing.Color get_Crimson();
public static System.Drawing.Color get_Cyan();
public static System.Drawing.Color get_DarkBlue();
public static System.Drawing.Color get_DarkCyan();
public static System.Drawing.Color get_DarkGoldenrod();
public static System.Drawing.Color get_DarkGray();
public static System.Drawing.Color get_DarkGreen();
public static System.Drawing.Color get_DarkKhaki();
public static System.Drawing.Color get_DarkMagenta();
public static System.Drawing.Color get_DarkOliveGreen();
public static System.Drawing.Color get_DarkOrange();
public static System.Drawing.Color get_DarkOrchid();
public static System.Drawing.Color get_DarkRed();
public static System.Drawing.Color get_DarkSalmon();
public static System.Drawing.Color get_DarkSeaGreen();
public static System.Drawing.Color get_DarkSlateBlue();
public static System.Drawing.Color get_DarkSlateGray();
public static System.Drawing.Color get_DarkTurquoise();
public static System.Drawing.Color get_DarkViolet();
public static System.Drawing.Color get_DeepPink();
public static System.Drawing.Color get_DeepSkyBlue();
public static System.Drawing.Color get_DimGray();
```

```
public static System.Drawing.Color get_DodgerBlue();
public static System.Drawing.Color get_Firebrick();
public static System.Drawing.Color get_FloralWhite();
public static System.Drawing.Color get_ForestGreen();
public static System.Drawing.Color get_Fuchsia();
public ubyte get_G();
public static System.Drawing.Color get_Gainsboro();
public static System.Drawing.Color get_GhostWhite();
public static System.Drawing.Color get_Gold();
public static System.Drawing.Color get_Goldenrod();
public static System.Drawing.Color get_Gray();
public static System.Drawing.Color get_Green();
public static System.Drawing.Color get_GreenYellow();
public static System.Drawing.Color get_Honeydew();
public static System.Drawing.Color get_HotPink();
public static System.Drawing.Color get_IndianRed();
public static System.Drawing.Color get_Indigo();
public boolean get_IsEmpty();
public boolean get_IsKnownColor();
public boolean get_IsNamedColor();
public boolean get_IsSystemColor();
public static System.Drawing.Color get_Ivory();
public static System.Drawing.Color get_Khaki();
public static System.Drawing.Color get_Lavender();
public static System.Drawing.Color get_LavenderBlush();
public static System.Drawing.Color get_LawnGreen();
public static System.Drawing.Color get_LemonChiffon();
public static System.Drawing.Color get_LightBlue();
public static System.Drawing.Color get_LightCoral();
public static System.Drawing.Color get_LightCyan();
public static System.Drawing.Color get_LightGoldenrodYellow();
public static System.Drawing.Color get_LightGray();
public static System.Drawing.Color get_LightGreen();
public static System.Drawing.Color get_LightPink();
public static System.Drawing.Color get_LightSalmon();
public static System.Drawing.Color get_LightSeaGreen();
public static System.Drawing.Color get_LightSkyBlue();
public static System.Drawing.Color get_LightSlateGray();
public static System.Drawing.Color get_LightSteelBlue();
public static System.Drawing.Color get_LightYellow();
public static System.Drawing.Color get_Lime();
public static System.Drawing.Color get_LimeGreen();
public static System.Drawing.Color get_Linen();
public static System.Drawing.Color get_Magenta();
public static System.Drawing.Color get_Maroon();
public static System.Drawing.Color get_MediumAquamarine();
public static System.Drawing.Color get_MediumBlue();
public static System.Drawing.Color get_MediumOrchid();
public static System.Drawing.Color get_MediumPurple();
public static System.Drawing.Color get_MediumSeaGreen();
public static System.Drawing.Color get_MediumSlateBlue();
public static System.Drawing.Color get_MediumSpringGreen();
public static System.Drawing.Color get_MediumTurquoise();
public static System.Drawing.Color get_MediumVioletRed();
public static System.Drawing.Color get_MidnightBlue();
public static System.Drawing.Color get_MintCream();
public static System.Drawing.Color get_MistyRose();
public static System.Drawing.Color get_Moccasin();
public System.String get_Name();
public static System.Drawing.Color get_NavajoWhite();
public static System.Drawing.Color get_Navy();
public static System.Drawing.Color get_OldLace();
public static System.Drawing.Color get_Olive();
public static System.Drawing.Color get_OliveDrab();
public static System.Drawing.Color get_Orange();
public static System.Drawing.Color get_OrangeRed();
public static System.Drawing.Color get_Orchid();
public static System.Drawing.Color get_PaleGoldenrod();
```

```
public static System.Drawing.Color get_PaleGreen();
public static System.Drawing.Color get_PaleTurquoise();
public static System.Drawing.Color get_PaleVioletRed();
public static System.Drawing.Color get_PapayaWhip();
public static System.Drawing.Color get_PeachPuff();
public static System.Drawing.Color get_Peru();
public static System.Drawing.Color get_Pink();
public static System.Drawing.Color get_Plum();
public static System.Drawing.Color get_PowderBlue();
public static System.Drawing.Color get_Purple();
public ubyte get_R();
public static System.Drawing.Color get_Red();
public static System.Drawing.Color get_RosyBrown();
public static System.Drawing.Color get_RoyalBlue();
public static System.Drawing.Color get_SaddleBrown();
public static System.Drawing.Color get_Salmon();
public static System.Drawing.Color get_SandyBrown();
public static System.Drawing.Color get_SeaGreen();
public static System.Drawing.Color get_SeaShell();
public static System.Drawing.Color get_Sienna();
public static System.Drawing.Color get_Silver();
public static System.Drawing.Color get_SkyBlue();
public static System.Drawing.Color get_SlateBlue();
public static System.Drawing.Color get_SlateGray();
public static System.Drawing.Color get_Snow();
public static System.Drawing.Color get_SpringGreen();
public static System.Drawing.Color get_SteelBlue();
public static System.Drawing.Color get_Tan();
public static System.Drawing.Color get_Teal();
public static System.Drawing.Color get_Thistle();
public static System.Drawing.Color get_Tomato();
public static System.Drawing.Color get_Transparent();
public static System.Drawing.Color get_Turquoise();
public static System.Drawing.Color get_Violet();
public static System.Drawing.Color get_Wheat();
public static System.Drawing.Color get_White();
public static System.Drawing.Color get_WhiteSmoke();
public static System.Drawing.Color get_Yellow();
public static System.Drawing.Color get_YellowGreen();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public static System.Drawing.Color FromArgb(
   int argb
);
public static System.Drawing.Color FromArgb(
   int alpha,
   System.Drawing.Color baseColor
);
public static System.Drawing.Color FromArgb(
   int red,
   int green,
   int blue
);
public static System.Drawing.Color FromArgb(
   int alpha,
   int red,
   int green,
   int blue
);
public static System.Drawing.Color FromKnownColor(
   System.Drawing.KnownColor color
);
```

```
public static System.Drawing.Color FromName(
   System.String name
);
public float GetBrightness();
public int GetHashCode();
public float GetHue();
public float GetSaturation();
public int ToArgb();
public System.Drawing.KnownColor ToKnownColor();
public System.String ToString();
```

Public Operators

```
public static boolean op_Equality(
   System.Drawing.Color left,
   System.Drawing.Color right
);
public static boolean op_Inequality(
   System.Drawing.Color left,
   System.Drawing.Color right
);
```

Public Fields

```
public static final System.Drawing.Color Empty;
```

ColorConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ColorConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public ColorConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

ColorTranslator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ColorTranslator
   extends System.Object
```

Public Methods

```
public static System.Drawing.Color FromHtml(
   System.String htmlColor
);
public static System.Drawing.Color FromOle(
   int oleColor
);
public static System.Drawing.Color FromWin32(
   int win32Color
);
public static System.String ToHtml(
   System.Drawing.Color c
);
public static int ToOle(
   System.Drawing.Color c
);
public static int ToWin32(
   System.Drawing.Color c
);
```

ContentAlignment Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Drawing.Design.ContentAlignmentEditor, System.Drawing.De
sign, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public enum ContentAlignment
```

Members

BottomCenter

BottomLeft

BottomRight

MiddleCenter

MiddleLeft

MiddleRight

TopCenter

TopLeft

TopRight

Font Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Drawi
ng.FontConverter, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Drawing.Design.FontEditor, System.Drawing.Design, Versio
n=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class Font
   extends System.MarshalByRefObject
   implements System.ICloneable, System.Runtime.Serialization.ISerializable, System.IDisposab
le
```

Public Constructors

```
public Font(
   System.Drawing.Font prototype,
   System.Drawing.FontStyle newStyle
);
public Font(
   System.Drawing.FontFamily family,
   float emSize
);
public Font(
   System.String familyName,
   float emSize
);
public Font(
   System.Drawing.FontFamily family,
   float emSize,
   System.Drawing.FontStyle style
);
public Font(
   System.Drawing.FontFamily family,
   float emSize,
   System.Drawing.GraphicsUnit unit
);
public Font(
   System.String familyName,
   float emSize,
   System.Drawing.FontStyle style
);
public Font(
   System.String familyName,
   float emSize,
   System.Drawing.GraphicsUnit unit
);
public Font(
   System.Drawing.FontFamily family,
   float emSize,
   System.Drawing.FontStyle style,
   System.Drawing.GraphicsUnit unit
);
public Font(
   System.String familyName,
   float emSize,
```

```
   System.Drawing.FontStyle style,
   System.Drawing.GraphicsUnit unit
);
public Font(
   System.Drawing.FontFamily family,
   float emSize,
   System.Drawing.FontStyle style,
   System.Drawing.GraphicsUnit unit,
   ubyte gdiCharSet
);
public Font(
   System.String familyName,
   float emSize,
   System.Drawing.FontStyle style,
   System.Drawing.GraphicsUnit unit,
   ubyte gdiCharSet
);
public Font(
   System.Drawing.FontFamily family,
   float emSize,
   System.Drawing.FontStyle style,
   System.Drawing.GraphicsUnit unit,
   ubyte gdiCharSet,
   boolean gdiVerticalFont
);
public Font(
   System.String familyName,
   float emSize,
   System.Drawing.FontStyle style,
   System.Drawing.GraphicsUnit unit,
   ubyte gdiCharSet,
   boolean gdiVerticalFont
);
```

Public Properties

```
public boolean get_Bold();
public System.Drawing.FontFamily get_FontFamily();
public ubyte get_GdiCharSet();
public boolean get_GdiVerticalFont();
public int get_Height();
public boolean get_Italic();
public System.String get_Name();
public float get_Size();
public float get_SizeInPoints();
public boolean get_Strikeout();
public System.Drawing.FontStyle get_Style();
public boolean get_Underline();
public System.Drawing.GraphicsUnit get_Unit();
```

Public Methods

```
public final System.Object Clone();
public final void Dispose();
public boolean Equals(
   System.Object obj
);
public static System.Drawing.Font FromHdc(
   System.IntPtr hdc
);
public static System.Drawing.Font FromHfont(
   System.IntPtr hfont
);
public static System.Drawing.Font FromLogFont(
```

```
   System.Object lf
);
public static System.Drawing.Font FromLogFont(
   System.Object lf,
   System.IntPtr hdc
);
public int GetHashCode();
public float GetHeight();
public float GetHeight(
   float dpi
);
public float GetHeight(
   System.Drawing.Graphics graphics
);
public System.IntPtr ToHfont();
public void ToLogFont(
   System.Object logFont
);
public void ToLogFont(
   System.Object logFont,
   System.Drawing.Graphics graphics
);
public System.String ToString();
```

Protected Methods

```
protected void Finalize();
```

FontConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FontConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public FontConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.Object CreateInstance(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Collections.IDictionary propertyValues
);
public boolean GetCreateInstanceSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value,
   System.Attribute[] attributes
);
public boolean GetPropertiesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

FontConverter.FontNameConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FontNameConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public FontNameConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

Protected Methods

```
protected void Finalize();
```

FontConverter.FontUnitConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FontUnitConverter
   extends System.ComponentModel.EnumConverter
```

Public Constructors

```
public FontUnitConverter();
```

Public Methods

```
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
```

FontFamily Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FontFamily
   extends System.MarshalByRefObject
   implements System.IDisposable
```

Public Constructors

```
public FontFamily(
   System.Drawing.Text.GenericFontFamilies genericFamily
);
public FontFamily(
   System.String name
);
public FontFamily(
   System.String name,
   System.Drawing.Text.FontCollection fontCollection
);
```

Public Properties

```
public static System.Drawing.FontFamily[] get_Families();
public static System.Drawing.FontFamily get_GenericMonospace();
public static System.Drawing.FontFamily get_GenericSansSerif();
public static System.Drawing.FontFamily get_GenericSerif();
public System.String get_Name();
```

Public Methods

```
public final void Dispose();
public boolean Equals(
   System.Object obj
);
public int GetCellAscent(
   System.Drawing.FontStyle style
);
public int GetCellDescent(
   System.Drawing.FontStyle style
);
public int GetEmHeight(
   System.Drawing.FontStyle style
);
public static System.Drawing.FontFamily[] GetFamilies(
   System.Drawing.Graphics graphics
);
public int GetHashCode();
public int GetLineSpacing(
   System.Drawing.FontStyle style
);
public System.String GetName(
   int language
);
public boolean IsStyleAvailable(
   System.Drawing.FontStyle style
);
```

```
public System.String ToString();
```

Protected Methods

```
protected void Finalize();
```

FontStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum FontStyle
```

Members

Bold

Italic

Regular

Strikeout

Underline

Graphics Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class Graphics
   extends System.MarshalByRefObject
   implements System.IDisposable
```

Public Properties

```
public System.Drawing.Region get_Clip();
public void set_Clip(
   System.Drawing.Region value
);
public System.Drawing.RectangleF get_ClipBounds();
public System.Drawing.Drawing2D.CompositingMode get_CompositingMode();
public void set_CompositingMode(
   System.Drawing.Drawing2D.CompositingMode value
);
public System.Drawing.Drawing2D.CompositingQuality get_CompositingQuality();
public void set_CompositingQuality(
   System.Drawing.Drawing2D.CompositingQuality value
);
public float get_DpiX();
public float get_DpiY();
public System.Drawing.Drawing2D.InterpolationMode get_InterpolationMode();
public void set_InterpolationMode(
   System.Drawing.Drawing2D.InterpolationMode value
);
public boolean get_IsClipEmpty();
public boolean get_IsVisibleClipEmpty();
public float get_PageScale();
public void set_PageScale(
   float value
);
public System.Drawing.GraphicsUnit get_PageUnit();
public void set_PageUnit(
   System.Drawing.GraphicsUnit value
);
public System.Drawing.Drawing2D.PixelOffsetMode get_PixelOffsetMode();
public void set_PixelOffsetMode(
   System.Drawing.Drawing2D.PixelOffsetMode value
);
public System.Drawing.Point get_RenderingOrigin();
public void set_RenderingOrigin(
   System.Drawing.Point value
);
public System.Drawing.Drawing2D.SmoothingMode get_SmoothingMode();
public void set_SmoothingMode(
   System.Drawing.Drawing2D.SmoothingMode value
);
public int get_TextContrast();
public void set_TextContrast(
   int value
);
public System.Drawing.Text.TextRenderingHint get_TextRenderingHint();
public void set_TextRenderingHint(
   System.Drawing.Text.TextRenderingHint value
);
public System.Drawing.Drawing2D.Matrix get_Transform();
```

```
public void set_Transform(
   System.Drawing.Drawing2D.Matrix value
);
public System.Drawing.RectangleF get_VisibleClipBounds();
```

Public Methods

```
public void AddMetafileComment(
   ubyte[] data
);
public System.Drawing.Drawing2D.GraphicsContainer BeginContainer();
public System.Drawing.Drawing2D.GraphicsContainer BeginContainer(
   System.Drawing.Rectangle dstrect,
   System.Drawing.Rectangle srcrect,
   System.Drawing.GraphicsUnit unit
);
public System.Drawing.Drawing2D.GraphicsContainer BeginContainer(
   System.Drawing.RectangleF dstrect,
   System.Drawing.RectangleF srcrect,
   System.Drawing.GraphicsUnit unit
);
public void Clear(
   System.Drawing.Color color
);
public final void Dispose();
public void DrawArc(
   System.Drawing.Pen pen,
   System.Drawing.Rectangle rect,
   float startAngle,
   float sweepAngle
);
public void DrawArc(
   System.Drawing.Pen pen,
   System.Drawing.RectangleF rect,
   float startAngle,
   float sweepAngle
);
public void DrawArc(
   System.Drawing.Pen pen,
   int x,
   int y,
   int width,
   int height,
   int startAngle,
   int sweepAngle
);
public void DrawArc(
   System.Drawing.Pen pen,
   float x,
   float y,
   float width,
   float height,
   float startAngle,
   float sweepAngle
);
public void DrawBezier(
   System.Drawing.Pen pen,
   System.Drawing.Point pt1,
   System.Drawing.Point pt2,
   System.Drawing.Point pt3,
   System.Drawing.Point pt4
);
public void DrawBezier(
   System.Drawing.Pen pen,
   System.Drawing.PointF pt1,
   System.Drawing.PointF pt2,
```

```
   System.Drawing.PointF pt3,
   System.Drawing.PointF pt4
);
public void DrawBezier(
   System.Drawing.Pen pen,
   float x1,
   float y1,
   float x2,
   float y2,
   float x3,
   float y3,
   float x4,
   float y4
);
public void DrawBeziers(
   System.Drawing.Pen pen,
   System.Drawing.PointF[] points
);
public void DrawBeziers(
   System.Drawing.Pen pen,
   System.Drawing.Point[] points
);
public void DrawClosedCurve(
   System.Drawing.Pen pen,
   System.Drawing.PointF[] points
);
public void DrawClosedCurve(
   System.Drawing.Pen pen,
   System.Drawing.Point[] points
);
public void DrawClosedCurve(
   System.Drawing.Pen pen,
   System.Drawing.PointF[] points,
   float tension,
   System.Drawing.Drawing2D.FillMode fillmode
);
public void DrawClosedCurve(
   System.Drawing.Pen pen,
   System.Drawing.Point[] points,
   float tension,
   System.Drawing.Drawing2D.FillMode fillmode
);
public void DrawCurve(
   System.Drawing.Pen pen,
   System.Drawing.PointF[] points
);
public void DrawCurve(
   System.Drawing.Pen pen,
   System.Drawing.Point[] points
);
public void DrawCurve(
   System.Drawing.Pen pen,
   System.Drawing.PointF[] points,
   float tension
);
public void DrawCurve(
   System.Drawing.Pen pen,
   System.Drawing.Point[] points,
   float tension
);
public void DrawCurve(
   System.Drawing.Pen pen,
   System.Drawing.PointF[] points,
   int offset,
   int numberOfSegments
);
public void DrawCurve(
   System.Drawing.Pen pen,
```

```
      System.Drawing.PointF[] points,
      int offset,
      int numberOfSegments,
      float tension
   );
   public void DrawCurve(
      System.Drawing.Pen pen,
      System.Drawing.Point[] points,
      int offset,
      int numberOfSegments,
      float tension
   );
   public void DrawEllipse(
      System.Drawing.Pen pen,
      System.Drawing.Rectangle rect
   );
   public void DrawEllipse(
      System.Drawing.Pen pen,
      System.Drawing.RectangleF rect
   );
   public void DrawEllipse(
      System.Drawing.Pen pen,
      int x,
      int y,
      int width,
      int height
   );
   public void DrawEllipse(
      System.Drawing.Pen pen,
      float x,
      float y,
      float width,
      float height
   );
   public void DrawIcon(
      System.Drawing.Icon icon,
      System.Drawing.Rectangle targetRect
   );
   public void DrawIcon(
      System.Drawing.Icon icon,
      int x,
      int y
   );
   public void DrawIconUnstretched(
      System.Drawing.Icon icon,
      System.Drawing.Rectangle targetRect
   );
   public void DrawImage(
      System.Drawing.Image image,
      System.Drawing.Point point
   );
   public void DrawImage(
      System.Drawing.Image image,
      System.Drawing.PointF point
   );
   public void DrawImage(
      System.Drawing.Image image,
      System.Drawing.PointF[] destPoints
   );
   public void DrawImage(
      System.Drawing.Image image,
      System.Drawing.Point[] destPoints
   );
   public void DrawImage(
      System.Drawing.Image image,
      System.Drawing.Rectangle rect
   );
   public void DrawImage(
```

```
   System.Drawing.Image image,
   System.Drawing.RectangleF rect
);
public void DrawImage(
   System.Drawing.Image image,
   int x,
   int y
);
public void DrawImage(
   System.Drawing.Image image,
   float x,
   float y
);
public void DrawImage(
   System.Drawing.Image image,
   System.Drawing.PointF[] destPoints,
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit srcUnit
);
public void DrawImage(
   System.Drawing.Image image,
   System.Drawing.Point[] destPoints,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit srcUnit
);
public void DrawImage(
   System.Drawing.Image image,
   System.Drawing.Rectangle destRect,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit srcUnit
);
public void DrawImage(
   System.Drawing.Image image,
   System.Drawing.RectangleF destRect,
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit srcUnit
);
public void DrawImage(
   System.Drawing.Image image,
   int x,
   int y,
   int width,
   int height
);
public void DrawImage(
   System.Drawing.Image image,
   int x,
   int y,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit srcUnit
);
public void DrawImage(
   System.Drawing.Image image,
   float x,
   float y,
   float width,
   float height
);
public void DrawImage(
   System.Drawing.Image image,
   float x,
   float y,
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit srcUnit
);
public void DrawImage(
   System.Drawing.Image image,
   System.Drawing.PointF[] destPoints,
```

```
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
public void DrawImage(
   System.Drawing.Image image,
   System.Drawing.Point[] destPoints,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
public void DrawImage(
   System.Drawing.Image image,
   System.Drawing.PointF[] destPoints,
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Imaging.ImageAttributes imageAttr,
   System.Drawing.Graphics+DrawImageAbort callback
);
public void DrawImage(
   System.Drawing.Image image,
   System.Drawing.Point[] destPoints,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Imaging.ImageAttributes imageAttr,
   System.Drawing.Graphics+DrawImageAbort callback
);
public void DrawImage(
   System.Drawing.Image image,
   System.Drawing.PointF[] destPoints,
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Imaging.ImageAttributes imageAttr,
   System.Drawing.Graphics+DrawImageAbort callback,
   int callbackData
);
public void DrawImage(
   System.Drawing.Image image,
   System.Drawing.Point[] destPoints,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Imaging.ImageAttributes imageAttr,
   System.Drawing.Graphics+DrawImageAbort callback,
   int callbackData
);
public void DrawImage(
   System.Drawing.Image image,
   System.Drawing.Rectangle destRect,
   int srcX,
   int srcY,
   int srcWidth,
   int srcHeight,
   System.Drawing.GraphicsUnit srcUnit
);
public void DrawImage(
   System.Drawing.Image image,
   System.Drawing.Rectangle destRect,
   float srcX,
   float srcY,
   float srcWidth,
   float srcHeight,
   System.Drawing.GraphicsUnit srcUnit
);
public void DrawImage(
   System.Drawing.Image image,
   System.Drawing.Rectangle destRect,
   int srcX,
   int srcY,
```

```
    int srcWidth,
    int srcHeight,
    System.Drawing.GraphicsUnit srcUnit,
    System.Drawing.Imaging.ImageAttributes imageAttr
);
public void DrawImage(
    System.Drawing.Image image,
    System.Drawing.Rectangle destRect,
    float srcX,
    float srcY,
    float srcWidth,
    float srcHeight,
    System.Drawing.GraphicsUnit srcUnit,
    System.Drawing.Imaging.ImageAttributes imageAttrs
);
public void DrawImage(
    System.Drawing.Image image,
    System.Drawing.Rectangle destRect,
    int srcX,
    int srcY,
    int srcWidth,
    int srcHeight,
    System.Drawing.GraphicsUnit srcUnit,
    System.Drawing.Imaging.ImageAttributes imageAttr,
    System.Drawing.Graphics+DrawImageAbort callback
);
public void DrawImage(
    System.Drawing.Image image,
    System.Drawing.Rectangle destRect,
    float srcX,
    float srcY,
    float srcWidth,
    float srcHeight,
    System.Drawing.GraphicsUnit srcUnit,
    System.Drawing.Imaging.ImageAttributes imageAttrs,
    System.Drawing.Graphics+DrawImageAbort callback
);
public void DrawImage(
    System.Drawing.Image image,
    System.Drawing.Rectangle destRect,
    int srcX,
    int srcY,
    int srcWidth,
    int srcHeight,
    System.Drawing.GraphicsUnit srcUnit,
    System.Drawing.Imaging.ImageAttributes imageAttrs,
    System.Drawing.Graphics+DrawImageAbort callback,
    System.IntPtr callbackData
);
public void DrawImage(
    System.Drawing.Image image,
    System.Drawing.Rectangle destRect,
    float srcX,
    float srcY,
    float srcWidth,
    float srcHeight,
    System.Drawing.GraphicsUnit srcUnit,
    System.Drawing.Imaging.ImageAttributes imageAttrs,
    System.Drawing.Graphics+DrawImageAbort callback,
    System.IntPtr callbackData
);
public void DrawImageUnscaled(
    System.Drawing.Image image,
    System.Drawing.Point point
);
public void DrawImageUnscaled(
    System.Drawing.Image image,
    System.Drawing.Rectangle rect
```

```
);
public void DrawImageUnscaled(
   System.Drawing.Image image,
   int x,
   int y
);
public void DrawImageUnscaled(
   System.Drawing.Image image,
   int x,
   int y,
   int width,
   int height
);
public void DrawLine(
   System.Drawing.Pen pen,
   System.Drawing.Point pt1,
   System.Drawing.Point pt2
);
public void DrawLine(
   System.Drawing.Pen pen,
   System.Drawing.PointF pt1,
   System.Drawing.PointF pt2
);
public void DrawLine(
   System.Drawing.Pen pen,
   int x1,
   int y1,
   int x2,
   int y2
);
public void DrawLine(
   System.Drawing.Pen pen,
   float x1,
   float y1,
   float x2,
   float y2
);
public void DrawLines(
   System.Drawing.Pen pen,
   System.Drawing.PointF[] points
);
public void DrawLines(
   System.Drawing.Pen pen,
   System.Drawing.Point[] points
);
public void DrawPath(
   System.Drawing.Pen pen,
   System.Drawing.Drawing2D.GraphicsPath path
);
public void DrawPie(
   System.Drawing.Pen pen,
   System.Drawing.Rectangle rect,
   float startAngle,
   float sweepAngle
);
public void DrawPie(
   System.Drawing.Pen pen,
   System.Drawing.RectangleF rect,
   float startAngle,
   float sweepAngle
);
public void DrawPie(
   System.Drawing.Pen pen,
   int x,
   int y,
   int width,
   int height,
   int startAngle,
```

```
    int sweepAngle
);
public void DrawPie(
    System.Drawing.Pen pen,
    float x,
    float y,
    float width,
    float height,
    float startAngle,
    float sweepAngle
);
public void DrawPolygon(
    System.Drawing.Pen pen,
    System.Drawing.PointF[] points
);
public void DrawPolygon(
    System.Drawing.Pen pen,
    System.Drawing.Point[] points
);
public void DrawRectangle(
    System.Drawing.Pen pen,
    System.Drawing.Rectangle rect
);
public void DrawRectangle(
    System.Drawing.Pen pen,
    int x,
    int y,
    int width,
    int height
);
public void DrawRectangle(
    System.Drawing.Pen pen,
    float x,
    float y,
    float width,
    float height
);
public void DrawRectangles(
    System.Drawing.Pen pen,
    System.Drawing.RectangleF[] rects
);
public void DrawRectangles(
    System.Drawing.Pen pen,
    System.Drawing.Rectangle[] rects
);
public void DrawString(
    System.String s,
    System.Drawing.Font font,
    System.Drawing.Brush brush,
    System.Drawing.PointF point
);
public void DrawString(
    System.String s,
    System.Drawing.Font font,
    System.Drawing.Brush brush,
    System.Drawing.RectangleF layoutRectangle
);
public void DrawString(
    System.String s,
    System.Drawing.Font font,
    System.Drawing.Brush brush,
    float x,
    float y
);
public void DrawString(
    System.String s,
    System.Drawing.Font font,
    System.Drawing.Brush brush,
```

```
   System.Drawing.PointF point,
   System.Drawing.StringFormat format
);
public void DrawString(
   System.String s,
   System.Drawing.Font font,
   System.Drawing.Brush brush,
   System.Drawing.RectangleF layoutRectangle,
   System.Drawing.StringFormat format
);
public void DrawString(
   System.String s,
   System.Drawing.Font font,
   System.Drawing.Brush brush,
   float x,
   float y,
   System.Drawing.StringFormat format
);
public void EndContainer(
   System.Drawing.Drawing2D.GraphicsContainer container
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Point destPoint,
   System.Drawing.Graphics+EnumerateMetafileProc callback
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.PointF destPoint,
   System.Drawing.Graphics+EnumerateMetafileProc callback
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.PointF[] destPoints,
   System.Drawing.Graphics+EnumerateMetafileProc callback
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Point[] destPoints,
   System.Drawing.Graphics+EnumerateMetafileProc callback
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Rectangle destRect,
   System.Drawing.Graphics+EnumerateMetafileProc callback
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.RectangleF destRect,
   System.Drawing.Graphics+EnumerateMetafileProc callback
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Point destPoint,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.PointF destPoint,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.PointF[] destPoints,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData
```

```
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Point[] destPoints,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Rectangle destRect,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.RectangleF destRect,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Point destPoint,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Point destPoint,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Graphics+EnumerateMetafileProc callback
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.PointF destPoint,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.PointF destPoint,
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Graphics+EnumerateMetafileProc callback
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.PointF[] destPoints,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.PointF[] destPoints,
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Graphics+EnumerateMetafileProc callback
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Point[] destPoints,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
```

```
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Point[] destPoints,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Graphics+EnumerateMetafileProc callback
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Rectangle destRect,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Rectangle destRect,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Graphics+EnumerateMetafileProc callback
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.RectangleF destRect,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.RectangleF destRect,
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Graphics+EnumerateMetafileProc callback
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Point destPoint,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.PointF destPoint,
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.PointF[] destPoints,
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Point[] destPoints,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData
);
public void EnumerateMetafile(
```

```
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Rectangle destRect,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.RectangleF destRect,
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit srcUnit,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Point destPoint,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit unit,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.PointF destPoint,
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit unit,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.PointF[] destPoints,
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit unit,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Point[] destPoints,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit unit,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.Rectangle destRect,
   System.Drawing.Rectangle srcRect,
   System.Drawing.GraphicsUnit unit,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
public void EnumerateMetafile(
   System.Drawing.Imaging.Metafile metafile,
   System.Drawing.RectangleF destRect,
   System.Drawing.RectangleF srcRect,
   System.Drawing.GraphicsUnit unit,
   System.Drawing.Graphics+EnumerateMetafileProc callback,
   System.IntPtr callbackData,
   System.Drawing.Imaging.ImageAttributes imageAttr
```

```
);
public void ExcludeClip(
   System.Drawing.Rectangle rect
);
public void ExcludeClip(
   System.Drawing.Region region
);
public void FillClosedCurve(
   System.Drawing.Brush brush,
   System.Drawing.PointF[] points
);
public void FillClosedCurve(
   System.Drawing.Brush brush,
   System.Drawing.Point[] points
);
public void FillClosedCurve(
   System.Drawing.Brush brush,
   System.Drawing.PointF[] points,
   System.Drawing.Drawing2D.FillMode fillmode
);
public void FillClosedCurve(
   System.Drawing.Brush brush,
   System.Drawing.Point[] points,
   System.Drawing.Drawing2D.FillMode fillmode
);
public void FillClosedCurve(
   System.Drawing.Brush brush,
   System.Drawing.PointF[] points,
   System.Drawing.Drawing2D.FillMode fillmode,
   float tension
);
public void FillClosedCurve(
   System.Drawing.Brush brush,
   System.Drawing.Point[] points,
   System.Drawing.Drawing2D.FillMode fillmode,
   float tension
);
public void FillEllipse(
   System.Drawing.Brush brush,
   System.Drawing.Rectangle rect
);
public void FillEllipse(
   System.Drawing.Brush brush,
   System.Drawing.RectangleF rect
);
public void FillEllipse(
   System.Drawing.Brush brush,
   int x,
   int y,
   int width,
   int height
);
public void FillEllipse(
   System.Drawing.Brush brush,
   float x,
   float y,
   float width,
   float height
);
public void FillPath(
   System.Drawing.Brush brush,
   System.Drawing.Drawing2D.GraphicsPath path
);
public void FillPie(
   System.Drawing.Brush brush,
   System.Drawing.Rectangle rect,
   float startAngle,
   float sweepAngle
```

```
);
public void FillPie(
   System.Drawing.Brush brush,
   int x,
   int y,
   int width,
   int height,
   int startAngle,
   int sweepAngle
);
public void FillPie(
   System.Drawing.Brush brush,
   float x,
   float y,
   float width,
   float height,
   float startAngle,
   float sweepAngle
);
public void FillPolygon(
   System.Drawing.Brush brush,
   System.Drawing.PointF[] points
);
public void FillPolygon(
   System.Drawing.Brush brush,
   System.Drawing.Point[] points
);
public void FillPolygon(
   System.Drawing.Brush brush,
   System.Drawing.PointF[] points,
   System.Drawing.Drawing2D.FillMode fillMode
);
public void FillPolygon(
   System.Drawing.Brush brush,
   System.Drawing.Point[] points,
   System.Drawing.Drawing2D.FillMode fillMode
);
public void FillRectangle(
   System.Drawing.Brush brush,
   System.Drawing.Rectangle rect
);
public void FillRectangle(
   System.Drawing.Brush brush,
   System.Drawing.RectangleF rect
);
public void FillRectangle(
   System.Drawing.Brush brush,
   int x,
   int y,
   int width,
   int height
);
public void FillRectangle(
   System.Drawing.Brush brush,
   float x,
   float y,
   float width,
   float height
);
public void FillRectangles(
   System.Drawing.Brush brush,
   System.Drawing.RectangleF[] rects
);
public void FillRectangles(
   System.Drawing.Brush brush,
   System.Drawing.Rectangle[] rects
);
public void FillRegion(
```

```
   System.Drawing.Brush brush,
   System.Drawing.Region region
);
public void Flush();
public void Flush(
   System.Drawing.Drawing2D.FlushIntention intention
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public static System.Drawing.Graphics FromHdc(
   System.IntPtr hdc
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public static System.Drawing.Graphics FromHdc(
   System.IntPtr hdc,
   System.IntPtr hdevice
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public static System.Drawing.Graphics FromHdcInternal(
   System.IntPtr hdc
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public static System.Drawing.Graphics FromHwnd(
   System.IntPtr hwnd
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public static System.Drawing.Graphics FromHwndInternal(
   System.IntPtr hwnd
);
public static System.Drawing.Graphics FromImage(
   System.Drawing.Image image
);
public static System.IntPtr GetHalftonePalette();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public System.IntPtr GetHdc();
public System.Drawing.Color GetNearestColor(
   System.Drawing.Color color
);
public void IntersectClip(
   System.Drawing.Rectangle rect
);
public void IntersectClip(
   System.Drawing.RectangleF rect
);
public void IntersectClip(
   System.Drawing.Region region
);
public boolean IsVisible(
   System.Drawing.Point point
);
public boolean IsVisible(
   System.Drawing.PointF point
);
public boolean IsVisible(
   System.Drawing.Rectangle rect
);
public boolean IsVisible(
   System.Drawing.RectangleF rect
);
public boolean IsVisible(
   int x,
   int y
);
public boolean IsVisible(
   float x,
   float y
);
public boolean IsVisible(
   int x,
```

```
   int y,
   int width,
   int height
);
public boolean IsVisible(
   float x,
   float y,
   float width,
   float height
);
public System.Drawing.Region[] MeasureCharacterRanges(
   System.String text,
   System.Drawing.Font font,
   System.Drawing.RectangleF layoutRect,
   System.Drawing.StringFormat stringFormat
);
public System.Drawing.SizeF MeasureString(
   System.String text,
   System.Drawing.Font font
);
public System.Drawing.SizeF MeasureString(
   System.String text,
   System.Drawing.Font font,
   int width
);
public System.Drawing.SizeF MeasureString(
   System.String text,
   System.Drawing.Font font,
   System.Drawing.SizeF layoutArea
);
public System.Drawing.SizeF MeasureString(
   System.String text,
   System.Drawing.Font font,
   int width,
   System.Drawing.StringFormat format
);
public System.Drawing.SizeF MeasureString(
   System.String text,
   System.Drawing.Font font,
   System.Drawing.PointF origin,
   System.Drawing.StringFormat stringFormat
);
public System.Drawing.SizeF MeasureString(
   System.String text,
   System.Drawing.Font font,
   System.Drawing.SizeF layoutArea,
   System.Drawing.StringFormat stringFormat
);
public System.Drawing.SizeF MeasureString(
   System.String text,
   System.Drawing.Font font,
   System.Drawing.SizeF layoutArea,
   System.Drawing.StringFormat stringFormat,
   int charactersFitted,
   int linesFilled
);
public void MultiplyTransform(
   System.Drawing.Drawing2D.Matrix matrix
);
public void MultiplyTransform(
   System.Drawing.Drawing2D.Matrix matrix,
   System.Drawing.Drawing2D.MatrixOrder order
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public void ReleaseHdc(
   System.IntPtr hdc
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
```

```
public void ReleaseHdcInternal(
   System.IntPtr hdc
);
public void ResetClip();
public void ResetTransform();
public void Restore(
   System.Drawing.Drawing2D.GraphicsState gstate
);
public void RotateTransform(
   float angle
);
public void RotateTransform(
   float angle,
   System.Drawing.Drawing2D.MatrixOrder order
);
public System.Drawing.Drawing2D.GraphicsState Save();
public void ScaleTransform(
   float sx,
   float sy
);
public void ScaleTransform(
   float sx,
   float sy,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void SetClip(
   System.Drawing.Drawing2D.GraphicsPath path
);
public void SetClip(
   System.Drawing.Graphics g
);
public void SetClip(
   System.Drawing.Rectangle rect
);
public void SetClip(
   System.Drawing.RectangleF rect
);
public void SetClip(
   System.Drawing.Drawing2D.GraphicsPath path,
   System.Drawing.Drawing2D.CombineMode combineMode
);
public void SetClip(
   System.Drawing.Graphics g,
   System.Drawing.Drawing2D.CombineMode combineMode
);
public void SetClip(
   System.Drawing.Rectangle rect,
   System.Drawing.Drawing2D.CombineMode combineMode
);
public void SetClip(
   System.Drawing.RectangleF rect,
   System.Drawing.Drawing2D.CombineMode combineMode
);
public void SetClip(
   System.Drawing.Region region,
   System.Drawing.Drawing2D.CombineMode combineMode
);
public void TransformPoints(
   System.Drawing.Drawing2D.CoordinateSpace destSpace,
   System.Drawing.Drawing2D.CoordinateSpace srcSpace,
   System.Drawing.PointF[] pts
);
public void TransformPoints(
   System.Drawing.Drawing2D.CoordinateSpace destSpace,
   System.Drawing.Drawing2D.CoordinateSpace srcSpace,
   System.Drawing.Point[] pts
);
public void TranslateClip(
```

```
      int dx,
      int dy
   );
   public void TranslateClip(
      float dx,
      float dy
   );
   public void TranslateTransform(
      float dx,
      float dy
   );
   public void TranslateTransform(
      float dx,
      float dy,
      System.Drawing.Drawing2D.MatrixOrder order
   );
```

Protected Methods

```
   protected void Finalize();
```

Graphics.DrawImageAbort Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
boolean DrawImageAbort(
   System.IntPtr callbackdata
);
```

Graphics.EnumerateMetafileProc Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
boolean EnumerateMetafileProc(
   System.Drawing.Imaging.EmfPlusRecordType recordType,
   int flags,
   int dataSize,
   System.IntPtr data,
   System.Drawing.Imaging.PlayRecordCallback callbackData
);
```

GraphicsUnit Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum GraphicsUnit
```

Members

Display

Document

Inch

Millimeter

Pixel

Point

World

Icon Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Drawing.Design.IconEditor, System.Drawing.Design, Versio
n=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Drawi
ng.IconConverter, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class Icon
   extends System.MarshalByRefObject
   implements System.Runtime.Serialization.ISerializable, System.ICloneable, System.IDisposab
le
```

Public Constructors

```
public Icon(
   System.IO.Stream stream
);
public Icon(
   System.String fileName
);
public Icon(
   System.Drawing.Icon original,
   System.Drawing.Size size
);
public Icon(
   System.Type type,
   System.String resource
);
public Icon(
   System.Drawing.Icon original,
   int width,
   int height
);
public Icon(
   System.IO.Stream stream,
   int width,
   int height
);
```

Public Properties

```
public System.IntPtr get_Handle();
public int get_Height();
public System.Drawing.Size get_Size();
public int get_Width();
```

Public Methods

```
public final System.Object Clone();
public final void Dispose();
public static System.Drawing.Icon FromHandle(
```

```
   System.IntPtr handle
);
public void Save(
   System.IO.Stream outputStream
);
public System.Drawing.Bitmap ToBitmap();
public System.String ToString();
```

Protected Methods

```
protected void Finalize();
```

IconConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class IconConverter
   extends System.ComponentModel.ExpandableObjectConverter
```

Public Constructors

```
public IconConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

Image Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.ImmutableObjectAttribute(Immutable = true) */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Drawing.Design.ImageEditor, System.Drawing.Design, Versi
on=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Drawi
ng.ImageConverter, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f
7f11d50a3a") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public abstract class Image
   extends System.MarshalByRefObject
   implements System.Runtime.Serialization.ISerializable, System.ICloneable, System.IDisposab
le
```

Public Properties

```
public int get_Flags();
public System.Guid[] get_FrameDimensionsList();
public int get_Height();
public float get_HorizontalResolution();
public System.Drawing.Imaging.ColorPalette get_Palette();
public void set_Palette(
   System.Drawing.Imaging.ColorPalette value
);
public System.Drawing.SizeF get_PhysicalDimension();
public System.Drawing.Imaging.PixelFormat get_PixelFormat();
public int[] get_PropertyIdList();
public System.Drawing.Imaging.PropertyItem[] get_PropertyItems();
public System.Drawing.Imaging.ImageFormat get_RawFormat();
public System.Drawing.Size get_Size();
public float get_VerticalResolution();
public int get_Width();
```

Public Methods

```
public final System.Object Clone();
public final void Dispose();
public static System.Drawing.Image FromFile(
   System.String filename
);
public static System.Drawing.Image FromFile(
   System.String filename,
   boolean useEmbeddedColorManagement
);
public static System.Drawing.Bitmap FromHbitmap(
   System.IntPtr hbitmap
);
public static System.Drawing.Bitmap FromHbitmap(
   System.IntPtr hbitmap,
   System.IntPtr hpalette
);
public static System.Drawing.Image FromStream(
   System.IO.Stream stream
);
```

```
public static System.Drawing.Image FromStream(
   System.IO.Stream stream,
   boolean useEmbeddedColorManagement
);
public System.Drawing.RectangleF GetBounds(
   System.Drawing.GraphicsUnit& pageUnit
);
public System.Drawing.Imaging.EncoderParameters GetEncoderParameterList(
   System.Guid encoder
);
public int GetFrameCount(
   System.Drawing.Imaging.FrameDimension dimension
);
public static int GetPixelFormatSize(
   System.Drawing.Imaging.PixelFormat pixfmt
);
public System.Drawing.Imaging.PropertyItem GetPropertyItem(
   int propid
);
public System.Drawing.Image GetThumbnailImage(
   int thumbWidth,
   int thumbHeight,
   System.Drawing.Image+GetThumbnailImageAbort callback,
   System.IntPtr callbackData
);
public static boolean IsAlphaPixelFormat(
   System.Drawing.Imaging.PixelFormat pixfmt
);
public static boolean IsCanonicalPixelFormat(
   System.Drawing.Imaging.PixelFormat pixfmt
);
public static boolean IsExtendedPixelFormat(
   System.Drawing.Imaging.PixelFormat pixfmt
);
public void RemovePropertyItem(
   int propid
);
public void RotateFlip(
   System.Drawing.RotateFlipType rotateFlipType
);
public void Save(
   System.String filename
);
public void Save(
   System.IO.Stream stream,
   System.Drawing.Imaging.ImageFormat format
);
public void Save(
   System.String filename,
   System.Drawing.Imaging.ImageFormat format
);
public void Save(
   System.IO.Stream stream,
   System.Drawing.Imaging.ImageCodecInfo encoder,
   System.Drawing.Imaging.EncoderParameters encoderParams
);
public void Save(
   System.String filename,
   System.Drawing.Imaging.ImageCodecInfo encoder,
   System.Drawing.Imaging.EncoderParameters encoderParams
);
public void SaveAdd(
   System.Drawing.Imaging.EncoderParameters encoderParams
);
public void SaveAdd(
   System.Drawing.Image image,
   System.Drawing.Imaging.EncoderParameters encoderParams
);
```

```
public int SelectActiveFrame(
   System.Drawing.Imaging.FrameDimension dimension,
   int frameIndex
);
public void SetPropertyItem(
   System.Drawing.Imaging.PropertyItem propitem
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

Image.GetThumbnailImageAbort Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
boolean GetThumbnailImageAbort();
```

ImageAnimator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ImageAnimator
   extends System.Object
```

Public Methods

```
public static void Animate(
   System.Drawing.Image image,
   System.EventHandler onFrameChangedHandler
);
public static boolean CanAnimate(
   System.Drawing.Image image
);
public static void StopAnimate(
   System.Drawing.Image image,
   System.EventHandler onFrameChangedHandler
);
public static void UpdateFrames();
public static void UpdateFrames(
   System.Drawing.Image image
);
```

ImageConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ImageConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public ImageConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value,
   System.Attribute[] attributes
);
public boolean GetPropertiesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

ImageFormatConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ImageFormatConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public ImageFormatConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

KnownColor Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum KnownColor
```

Members

ActiveBorder

ActiveCaption

ActiveCaptionText

AliceBlue

AntiqueWhite

AppWorkspace

Aqua

Aquamarine

Azure

Beige

Bisque

Black

BlanchedAlmond

Blue

BlueViolet

Brown

BurlyWood

CadetBlue

Chartreuse

Chocolate

Control

ControlDark

ControlDarkDark

ControlLight

ControlLightLight

ControlText

Coral

CornflowerBlue

Cornsilk

Crimson

Cyan

DarkBlue

DarkCyan

DarkGoldenrod

DarkGray

DarkGreen

DarkKhaki

DarkMagenta

DarkOliveGreen

DarkOrange

DarkOrchid

DarkRed

DarkSalmon

DarkSeaGreen

DarkSlateBlue

DarkSlateGray

DarkTurquoise

DarkViolet

DeepPink

DeepSkyBlue

Desktop

DimGray

DodgerBlue

Firebrick

FloralWhite

ForestGreen

Fuchsia

Gainsboro

GhostWhite

Gold

Goldenrod

Gray

GrayText

Green

GreenYellow

Highlight

HighlightText

Honeydew

HotPink

HotTrack

InactiveBorder

InactiveCaption

InactiveCaptionText

IndianRed

Indigo

Info

InfoText

Ivory

Khaki

Lavender

LavenderBlush

LawnGreen

LemonChiffon

LightBlue

LightCoral

LightCyan

LightGoldenrodYellow

LightGray

LightGreen

LightPink

LightSalmon

LightSeaGreen

LightSkyBlue

LightSlateGray

LightSteelBlue

LightYellow

Lime

LimeGreen

Linen

Magenta

Maroon

MediumAquamarine

MediumBlue

MediumOrchid

MediumPurple

MediumSeaGreen

MediumSlateBlue

MediumSpringGreen

MediumTurquoise

MediumVioletRed

Menu

MenuText

MidnightBlue

MintCream

MistyRose

Moccasin

NavajoWhite

Navy

OldLace

Olive

OliveDrab

Orange

OrangeRed

Orchid

PaleGoldenrod

PaleGreen

PaleTurquoise

PaleVioletRed

PapayaWhip

PeachPuff

Peru

Pink

Plum

PowderBlue

Purple

Red

RosyBrown

RoyalBlue

SaddleBrown

Salmon

SandyBrown

ScrollBar

SeaGreen

SeaShell

Sienna

Silver

SkyBlue

SlateBlue

SlateGray

Snow

SpringGreen

SteelBlue

Tan

Teal

Thistle

Tomato

Transparent

Turquoise

Violet

Wheat

White

WhiteSmoke

Window

WindowFrame

WindowText

Yellow

YellowGreen

Pen Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Pen
   extends System.MarshalByRefObject
   implements System.Drawing.Internal.ISystemColorTracker, System.ICloneable, System.IDisposa
ble
```

Public Constructors

```
public Pen(
   System.Drawing.Brush brush
);
public Pen(
   System.Drawing.Color color
);
public Pen(
   System.Drawing.Brush brush,
   float width
);
public Pen(
   System.Drawing.Color color,
   float width
);
```

Public Properties

```
public System.Drawing.Drawing2D.PenAlignment get_Alignment();
public void set_Alignment(
   System.Drawing.Drawing2D.PenAlignment value
);
public System.Drawing.Brush get_Brush();
public void set_Brush(
   System.Drawing.Brush value
);
public System.Drawing.Color get_Color();
public void set_Color(
   System.Drawing.Color value
);
public float[] get_CompoundArray();
public void set_CompoundArray(
   float[] value
);
public System.Drawing.Drawing2D.CustomLineCap get_CustomEndCap();
public void set_CustomEndCap(
   System.Drawing.Drawing2D.CustomLineCap value
);
public System.Drawing.Drawing2D.CustomLineCap get_CustomStartCap();
public void set_CustomStartCap(
   System.Drawing.Drawing2D.CustomLineCap value
);
public System.Drawing.Drawing2D.DashCap get_DashCap();
public void set_DashCap(
   System.Drawing.Drawing2D.DashCap value
);
public float get_DashOffset();
public void set_DashOffset(
   float value
```

```
);
public float[] get_DashPattern();
public void set_DashPattern(
   float[] value
);
public System.Drawing.Drawing2D.DashStyle get_DashStyle();
public void set_DashStyle(
   System.Drawing.Drawing2D.DashStyle value
);
public System.Drawing.Drawing2D.LineCap get_EndCap();
public void set_EndCap(
   System.Drawing.Drawing2D.LineCap value
);
public System.Drawing.Drawing2D.LineJoin get_LineJoin();
public void set_LineJoin(
   System.Drawing.Drawing2D.LineJoin value
);
public float get_MiterLimit();
public void set_MiterLimit(
   float value
);
public System.Drawing.Drawing2D.PenType get_PenType();
public System.Drawing.Drawing2D.LineCap get_StartCap();
public void set_StartCap(
   System.Drawing.Drawing2D.LineCap value
);
public System.Drawing.Drawing2D.Matrix get_Transform();
public void set_Transform(
   System.Drawing.Drawing2D.Matrix value
);
public float get_Width();
public void set_Width(
   float value
);
```

Public Methods

```
public final System.Object Clone();
public final void Dispose();
public void MultiplyTransform(
   System.Drawing.Drawing2D.Matrix matrix
);
public void MultiplyTransform(
   System.Drawing.Drawing2D.Matrix matrix,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void ResetTransform();
public void RotateTransform(
   float angle
);
public void RotateTransform(
   float angle,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void ScaleTransform(
   float sx,
   float sy
);
public void ScaleTransform(
   float sx,
   float sy,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void SetLineCap(
   System.Drawing.Drawing2D.LineCap startCap,
   System.Drawing.Drawing2D.LineCap endCap,
```

```
   System.Drawing.Drawing2D.DashCap dashCap
);
public void TranslateTransform(
   float dx,
   float dy
);
public void TranslateTransform(
   float dx,
   float dy,
   System.Drawing.Drawing2D.MatrixOrder order
);
```

Protected Methods

```
protected void Finalize();
```

Pens Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Pens
   extends System.Object
```

Public Properties

```
public static System.Drawing.Pen get_AliceBlue();
public static System.Drawing.Pen get_AntiqueWhite();
public static System.Drawing.Pen get_Aqua();
public static System.Drawing.Pen get_Aquamarine();
public static System.Drawing.Pen get_Azure();
public static System.Drawing.Pen get_Beige();
public static System.Drawing.Pen get_Bisque();
public static System.Drawing.Pen get_Black();
public static System.Drawing.Pen get_BlanchedAlmond();
public static System.Drawing.Pen get_Blue();
public static System.Drawing.Pen get_BlueViolet();
public static System.Drawing.Pen get_Brown();
public static System.Drawing.Pen get_BurlyWood();
public static System.Drawing.Pen get_CadetBlue();
public static System.Drawing.Pen get_Chartreuse();
public static System.Drawing.Pen get_Chocolate();
public static System.Drawing.Pen get_Coral();
public static System.Drawing.Pen get_CornflowerBlue();
public static System.Drawing.Pen get_Cornsilk();
public static System.Drawing.Pen get_Crimson();
public static System.Drawing.Pen get_Cyan();
public static System.Drawing.Pen get_DarkBlue();
public static System.Drawing.Pen get_DarkCyan();
public static System.Drawing.Pen get_DarkGoldenrod();
public static System.Drawing.Pen get_DarkGray();
public static System.Drawing.Pen get_DarkGreen();
public static System.Drawing.Pen get_DarkKhaki();
public static System.Drawing.Pen get_DarkMagenta();
public static System.Drawing.Pen get_DarkOliveGreen();
public static System.Drawing.Pen get_DarkOrange();
public static System.Drawing.Pen get_DarkOrchid();
public static System.Drawing.Pen get_DarkRed();
public static System.Drawing.Pen get_DarkSalmon();
public static System.Drawing.Pen get_DarkSeaGreen();
public static System.Drawing.Pen get_DarkSlateBlue();
public static System.Drawing.Pen get_DarkSlateGray();
public static System.Drawing.Pen get_DarkTurquoise();
public static System.Drawing.Pen get_DarkViolet();
public static System.Drawing.Pen get_DeepPink();
public static System.Drawing.Pen get_DeepSkyBlue();
public static System.Drawing.Pen get_DimGray();
public static System.Drawing.Pen get_DodgerBlue();
public static System.Drawing.Pen get_Firebrick();
public static System.Drawing.Pen get_FloralWhite();
public static System.Drawing.Pen get_ForestGreen();
public static System.Drawing.Pen get_Fuchsia();
public static System.Drawing.Pen get_Gainsboro();
public static System.Drawing.Pen get_GhostWhite();
public static System.Drawing.Pen get_Gold();
public static System.Drawing.Pen get_Goldenrod();
public static System.Drawing.Pen get_Gray();
public static System.Drawing.Pen get_Green();
```

```
public static System.Drawing.Pen get_GreenYellow();
public static System.Drawing.Pen get_Honeydew();
public static System.Drawing.Pen get_HotPink();
public static System.Drawing.Pen get_IndianRed();
public static System.Drawing.Pen get_Indigo();
public static System.Drawing.Pen get_Ivory();
public static System.Drawing.Pen get_Khaki();
public static System.Drawing.Pen get_Lavender();
public static System.Drawing.Pen get_LavenderBlush();
public static System.Drawing.Pen get_LawnGreen();
public static System.Drawing.Pen get_LemonChiffon();
public static System.Drawing.Pen get_LightBlue();
public static System.Drawing.Pen get_LightCoral();
public static System.Drawing.Pen get_LightCyan();
public static System.Drawing.Pen get_LightGoldenrodYellow();
public static System.Drawing.Pen get_LightGray();
public static System.Drawing.Pen get_LightGreen();
public static System.Drawing.Pen get_LightPink();
public static System.Drawing.Pen get_LightSalmon();
public static System.Drawing.Pen get_LightSeaGreen();
public static System.Drawing.Pen get_LightSkyBlue();
public static System.Drawing.Pen get_LightSlateGray();
public static System.Drawing.Pen get_LightSteelBlue();
public static System.Drawing.Pen get_LightYellow();
public static System.Drawing.Pen get_Lime();
public static System.Drawing.Pen get_LimeGreen();
public static System.Drawing.Pen get_Linen();
public static System.Drawing.Pen get_Magenta();
public static System.Drawing.Pen get_Maroon();
public static System.Drawing.Pen get_MediumAquamarine();
public static System.Drawing.Pen get_MediumBlue();
public static System.Drawing.Pen get_MediumOrchid();
public static System.Drawing.Pen get_MediumPurple();
public static System.Drawing.Pen get_MediumSeaGreen();
public static System.Drawing.Pen get_MediumSlateBlue();
public static System.Drawing.Pen get_MediumSpringGreen();
public static System.Drawing.Pen get_MediumTurquoise();
public static System.Drawing.Pen get_MediumVioletRed();
public static System.Drawing.Pen get_MidnightBlue();
public static System.Drawing.Pen get_MintCream();
public static System.Drawing.Pen get_MistyRose();
public static System.Drawing.Pen get_Moccasin();
public static System.Drawing.Pen get_NavajoWhite();
public static System.Drawing.Pen get_Navy();
public static System.Drawing.Pen get_OldLace();
public static System.Drawing.Pen get_Olive();
public static System.Drawing.Pen get_OliveDrab();
public static System.Drawing.Pen get_Orange();
public static System.Drawing.Pen get_OrangeRed();
public static System.Drawing.Pen get_Orchid();
public static System.Drawing.Pen get_PaleGoldenrod();
public static System.Drawing.Pen get_PaleGreen();
public static System.Drawing.Pen get_PaleTurquoise();
public static System.Drawing.Pen get_PaleVioletRed();
public static System.Drawing.Pen get_PapayaWhip();
public static System.Drawing.Pen get_PeachPuff();
public static System.Drawing.Pen get_Peru();
public static System.Drawing.Pen get_Pink();
public static System.Drawing.Pen get_Plum();
public static System.Drawing.Pen get_PowderBlue();
public static System.Drawing.Pen get_Purple();
public static System.Drawing.Pen get_Red();
public static System.Drawing.Pen get_RosyBrown();
public static System.Drawing.Pen get_RoyalBlue();
public static System.Drawing.Pen get_SaddleBrown();
public static System.Drawing.Pen get_Salmon();
public static System.Drawing.Pen get_SandyBrown();
public static System.Drawing.Pen get_SeaGreen();
```

```
public static System.Drawing.Pen get_SeaShell();
public static System.Drawing.Pen get_Sienna();
public static System.Drawing.Pen get_Silver();
public static System.Drawing.Pen get_SkyBlue();
public static System.Drawing.Pen get_SlateBlue();
public static System.Drawing.Pen get_SlateGray();
public static System.Drawing.Pen get_Snow();
public static System.Drawing.Pen get_SpringGreen();
public static System.Drawing.Pen get_SteelBlue();
public static System.Drawing.Pen get_Tan();
public static System.Drawing.Pen get_Teal();
public static System.Drawing.Pen get_Thistle();
public static System.Drawing.Pen get_Tomato();
public static System.Drawing.Pen get_Transparent();
public static System.Drawing.Pen get_Turquoise();
public static System.Drawing.Pen get_Violet();
public static System.Drawing.Pen get_Wheat();
public static System.Drawing.Pen get_White();
public static System.Drawing.Pen get_WhiteSmoke();
public static System.Drawing.Pen get_Yellow();
public static System.Drawing.Pen get_YellowGreen();
```

Point Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Drawi
ng.PointConverter, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f
7f11d50a3a") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class Point
   extends System.ValueType
```

Public Constructors

```
public Point(
   int dw
);
public Point(
   System.Drawing.Size sz
);
public Point(
   int x,
   int y
);
```

Public Properties

```
public boolean get_IsEmpty();
public int get_X();
public void set_X(
   int value
);
public int get_Y();
public void set_Y(
   int value
);
```

Public Methods

```
public static System.Drawing.Point Ceiling(
   System.Drawing.PointF value
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public void Offset(
   int dx,
   int dy
);
public static System.Drawing.Point Round(
   System.Drawing.PointF value
);
public System.String ToString();
public static System.Drawing.Point Truncate(
   System.Drawing.PointF value
);
```

Public Operators

```
public static System.Drawing.Point op_Addition(
   System.Drawing.Point pt,
   System.Drawing.Size sz
);
public static boolean op_Equality(
   System.Drawing.Point left,
   System.Drawing.Point right
);
public static System.Drawing.Size op_Explicit(
   System.Drawing.Point p
);
public static System.Drawing.PointF op_Implicit(
   System.Drawing.Point p
);
public static boolean op_Inequality(
   System.Drawing.Point left,
   System.Drawing.Point right
);
public static System.Drawing.Point op_Subtraction(
   System.Drawing.Point pt,
   System.Drawing.Size sz
);
```

Public Fields

```
public static final System.Drawing.Point Empty;
```

PointConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PointConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public PointConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.Object CreateInstance(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Collections.IDictionary propertyValues
);
public boolean GetCreateInstanceSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value,
   System.Attribute[] attributes
);
public boolean GetPropertiesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

PointF Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class PointF
   extends System.ValueType
```

Public Constructors

```
public PointF(
   float x,
   float y
);
```

Public Properties

```
public boolean get_IsEmpty();
public float get_X();
public void set_X(
   float value
);
public float get_Y();
public void set_Y(
   float value
);
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public System.String ToString();
```

Public Operators

```
public static System.Drawing.PointF op_Addition(
   System.Drawing.PointF pt,
   System.Drawing.Size sz
);
public static boolean op_Equality(
   System.Drawing.PointF left,
   System.Drawing.PointF right
);
public static boolean op_Inequality(
   System.Drawing.PointF left,
   System.Drawing.PointF right
);
public static System.Drawing.PointF op_Subtraction(
   System.Drawing.PointF pt,
   System.Drawing.Size sz
);
```

Public Fields

```
public static final System.Drawing.PointF Empty;
```

Rectangle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Drawi
ng.RectangleConverter, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b0
3f5f7f11d50a3a") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class Rectangle
   extends System.ValueType
```

Public Constructors

```
public Rectangle(
   System.Drawing.Point location,
   System.Drawing.Size size
);
public Rectangle(
   int x,
   int y,
   int width,
   int height
);
```

Public Properties

```
public int get_Bottom();
public int get_Height();
public void set_Height(
   int value
);
public boolean get_IsEmpty();
public int get_Left();
public System.Drawing.Point get_Location();
public void set_Location(
   System.Drawing.Point value
);
public int get_Right();
public System.Drawing.Size get_Size();
public void set_Size(
   System.Drawing.Size value
);
public int get_Top();
public int get_Width();
public void set_Width(
   int value
);
public int get_X();
public void set_X(
   int value
);
public int get_Y();
public void set_Y(
   int value
);
```

Public Methods

```
public static System.Drawing.Rectangle Ceiling(
   System.Drawing.RectangleF value
);
public boolean Contains(
   System.Drawing.Point pt
);
public boolean Contains(
   System.Drawing.Rectangle rect
);
public boolean Contains(
   int x,
   int y
);
public boolean Equals(
   System.Object obj
);
public static System.Drawing.Rectangle FromLTRB(
   int left,
   int top,
   int right,
   int bottom
);
public int GetHashCode();
public void Inflate(
   System.Drawing.Size size
);
public void Inflate(
   int width,
   int height
);
public static System.Drawing.Rectangle Inflate(
   System.Drawing.Rectangle rect,
   int x,
   int y
);
public void Intersect(
   System.Drawing.Rectangle rect
);
public static System.Drawing.Rectangle Intersect(
   System.Drawing.Rectangle a,
   System.Drawing.Rectangle b
);
public boolean IntersectsWith(
   System.Drawing.Rectangle rect
);
public void Offset(
   System.Drawing.Point pos
);
public void Offset(
   int x,
   int y
);
public static System.Drawing.Rectangle Round(
   System.Drawing.RectangleF value
);
public System.String ToString();
public static System.Drawing.Rectangle Truncate(
   System.Drawing.RectangleF value
);
public static System.Drawing.Rectangle Union(
   System.Drawing.Rectangle a,
   System.Drawing.Rectangle b
);
```

Public Operators

```
public static boolean op_Equality(
   System.Drawing.Rectangle left,
   System.Drawing.Rectangle right
);
public static boolean op_Inequality(
   System.Drawing.Rectangle left,
   System.Drawing.Rectangle right
);
```

Public Fields

```
public static final System.Drawing.Rectangle Empty;
```

RectangleConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RectangleConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public RectangleConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.Object CreateInstance(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Collections.IDictionary propertyValues
);
public boolean GetCreateInstanceSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value,
   System.Attribute[] attributes
);
public boolean GetPropertiesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

RectangleF Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RectangleF
   extends System.ValueType
```

Public Constructors

```
public RectangleF(
   System.Drawing.PointF location,
   System.Drawing.SizeF size
);
public RectangleF(
   float x,
   float y,
   float width,
   float height
);
```

Public Properties

```
public float get_Bottom();
public float get_Height();
public void set_Height(
   float value
);
public boolean get_IsEmpty();
public float get_Left();
public System.Drawing.PointF get_Location();
public void set_Location(
   System.Drawing.PointF value
);
public float get_Right();
public System.Drawing.SizeF get_Size();
public void set_Size(
   System.Drawing.SizeF value
);
public float get_Top();
public float get_Width();
public void set_Width(
   float value
);
public float get_X();
public void set_X(
   float value
);
public float get_Y();
public void set_Y(
   float value
);
```

Public Methods

```
public boolean Contains(
   System.Drawing.PointF pt
```

```
);
public boolean Contains(
   System.Drawing.RectangleF rect
);
public boolean Contains(
   float x,
   float y
);
public boolean Equals(
   System.Object obj
);
public static System.Drawing.RectangleF FromLTRB(
   float left,
   float top,
   float right,
   float bottom
);
public int GetHashCode();
public void Inflate(
   System.Drawing.SizeF size
);
public void Inflate(
   float x,
   float y
);
public static System.Drawing.RectangleF Inflate(
   System.Drawing.RectangleF rect,
   float x,
   float y
);
public void Intersect(
   System.Drawing.RectangleF rect
);
public static System.Drawing.RectangleF Intersect(
   System.Drawing.RectangleF a,
   System.Drawing.RectangleF b
);
public boolean IntersectsWith(
   System.Drawing.RectangleF rect
);
public void Offset(
   System.Drawing.PointF pos
);
public void Offset(
   float x,
   float y
);
public System.String ToString();
public static System.Drawing.RectangleF Union(
   System.Drawing.RectangleF a,
   System.Drawing.RectangleF b
);
```

Public Operators

```
public static boolean op_Equality(
   System.Drawing.RectangleF left,
   System.Drawing.RectangleF right
);
public static System.Drawing.RectangleF op_Implicit(
   System.Drawing.Rectangle r
);
public static boolean op_Inequality(
   System.Drawing.RectangleF left,
   System.Drawing.RectangleF right
);
```

Public Fields

```
public static final System.Drawing.RectangleF Empty;
```

Region Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class Region
   extends System.MarshalByRefObject
   implements System.IDisposable
```

Public Constructors

```
public Region();
public Region(
   System.Drawing.Drawing2D.GraphicsPath path
);
public Region(
   System.Drawing.Drawing2D.RegionData rgnData
);
public Region(
   System.Drawing.Rectangle rect
);
public Region(
   System.Drawing.RectangleF rect
);
```

Public Methods

```
public System.Drawing.Region Clone();
public void Complement(
   System.Drawing.Drawing2D.GraphicsPath path
);
public void Complement(
   System.Drawing.Rectangle rect
);
public void Complement(
   System.Drawing.RectangleF rect
);
public void Complement(
   System.Drawing.Region region
);
public final void Dispose();
public boolean Equals(
   System.Drawing.Region region,
   System.Drawing.Graphics g
);
public void Exclude(
   System.Drawing.Drawing2D.GraphicsPath path
);
public void Exclude(
   System.Drawing.Rectangle rect
);
public void Exclude(
   System.Drawing.RectangleF rect
);
public void Exclude(
   System.Drawing.Region region
);
public static System.Drawing.Region FromHrgn(
   System.IntPtr hrgn
```

```
);
public System.Drawing.RectangleF GetBounds(
   System.Drawing.Graphics g
);
public System.IntPtr GetHrgn(
   System.Drawing.Graphics g
);
public System.Drawing.Drawing2D.RegionData GetRegionData();
public System.Drawing.RectangleF[] GetRegionScans(
   System.Drawing.Drawing2D.Matrix matrix
);
public void Intersect(
   System.Drawing.Drawing2D.GraphicsPath path
);
public void Intersect(
   System.Drawing.Rectangle rect
);
public void Intersect(
   System.Drawing.RectangleF rect
);
public void Intersect(
   System.Drawing.Region region
);
public boolean IsEmpty(
   System.Drawing.Graphics g
);
public boolean IsInfinite(
   System.Drawing.Graphics g
);
public boolean IsVisible(
   System.Drawing.Point point
);
public boolean IsVisible(
   System.Drawing.PointF point
);
public boolean IsVisible(
   System.Drawing.Rectangle rect
);
public boolean IsVisible(
   System.Drawing.RectangleF rect
);
public boolean IsVisible(
   float x,
   float y
);
public boolean IsVisible(
   System.Drawing.Point point,
   System.Drawing.Graphics g
);
public boolean IsVisible(
   System.Drawing.PointF point,
   System.Drawing.Graphics g
);
public boolean IsVisible(
   System.Drawing.Rectangle rect,
   System.Drawing.Graphics g
);
public boolean IsVisible(
   System.Drawing.RectangleF rect,
   System.Drawing.Graphics g
);
public boolean IsVisible(
   int x,
   int y,
   System.Drawing.Graphics g
);
public boolean IsVisible(
   float x,
```

```
   float y,
   System.Drawing.Graphics g
);
public boolean IsVisible(
   int x,
   int y,
   int width,
   int height
);
public boolean IsVisible(
   float x,
   float y,
   float width,
   float height
);
public boolean IsVisible(
   int x,
   int y,
   int width,
   int height,
   System.Drawing.Graphics g
);
public boolean IsVisible(
   float x,
   float y,
   float width,
   float height,
   System.Drawing.Graphics g
);
public void MakeEmpty();
public void MakeInfinite();
public void Transform(
   System.Drawing.Drawing2D.Matrix matrix
);
public void Translate(
   int dx,
   int dy
);
public void Translate(
   float dx,
   float dy
);
public void Union(
   System.Drawing.Drawing2D.GraphicsPath path
);
public void Union(
   System.Drawing.Rectangle rect
);
public void Union(
   System.Drawing.RectangleF rect
);
public void Union(
   System.Drawing.Region region
);
public void Xor(
   System.Drawing.Drawing2D.GraphicsPath path
);
public void Xor(
   System.Drawing.Rectangle rect
);
public void Xor(
   System.Drawing.RectangleF rect
);
public void Xor(
   System.Drawing.Region region
);
```

Protected Methods

```
protected void Finalize();
```

RotateFlipType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum RotateFlipType
```

Members

Rotate180FlipNone

Rotate180FlipX

Rotate180FlipXY

Rotate180FlipY

Rotate270FlipNone

Rotate270FlipX

Rotate270FlipXY

Rotate270FlipY

Rotate90FlipNone

Rotate90FlipX

Rotate90FlipXY

Rotate90FlipY

RotateNoneFlipNone

RotateNoneFlipX

RotateNoneFlipXY

RotateNoneFlipY

Size Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Drawi
ng.SizeConverter, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a") */
public class Size
   extends System.ValueType
```

Public Constructors

```
public Size(
   System.Drawing.Point pt
);
public Size(
   int width,
   int height
);
```

Public Properties

```
public int get_Height();
public void set_Height(
   int value
);
public boolean get_IsEmpty();
public int get_Width();
public void set_Width(
   int value
);
```

Public Methods

```
public static System.Drawing.Size Ceiling(
   System.Drawing.SizeF value
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public static System.Drawing.Size Round(
   System.Drawing.SizeF value
);
public System.String ToString();
public static System.Drawing.Size Truncate(
   System.Drawing.SizeF value
);
```

Public Operators

```
public static System.Drawing.Size op_Addition(
   System.Drawing.Size sz1,
```

```
   System.Drawing.Size sz2
);
public static boolean op_Equality(
   System.Drawing.Size sz1,
   System.Drawing.Size sz2
);
public static System.Drawing.Point op_Explicit(
   System.Drawing.Size size
);
public static System.Drawing.SizeF op_Implicit(
   System.Drawing.Size p
);
public static boolean op_Inequality(
   System.Drawing.Size sz1,
   System.Drawing.Size sz2
);
public static System.Drawing.Size op_Subtraction(
   System.Drawing.Size sz1,
   System.Drawing.Size sz2
);
```

Public Fields

```
public static final System.Drawing.Size Empty;
```

SizeConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SizeConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public SizeConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.Object CreateInstance(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Collections.IDictionary propertyValues
);
public boolean GetCreateInstanceSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value,
   System.Attribute[] attributes
);
public boolean GetPropertiesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

SizeF Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class SizeF
   extends System.ValueType
```

Public Constructors

```
public SizeF(
   System.Drawing.PointF pt
);
public SizeF(
   System.Drawing.SizeF size
);
public SizeF(
   float width,
   float height
);
```

Public Properties

```
public float get_Height();
public void set_Height(
   float value
);
public boolean get_IsEmpty();
public float get_Width();
public void set_Width(
   float value
);
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public System.Drawing.PointF ToPointF();
public System.Drawing.Size ToSize();
public System.String ToString();
```

Public Operators

```
public static System.Drawing.SizeF op_Addition(
   System.Drawing.SizeF sz1,
   System.Drawing.SizeF sz2
);
public static boolean op_Equality(
   System.Drawing.SizeF sz1,
   System.Drawing.SizeF sz2
);
public static System.Drawing.PointF op_Explicit(
```

```
   System.Drawing.SizeF size
);
public static boolean op_Inequality(
   System.Drawing.SizeF sz1,
   System.Drawing.SizeF sz2
);
public static System.Drawing.SizeF op_Subtraction(
   System.Drawing.SizeF sz1,
   System.Drawing.SizeF sz2
);
```

Public Fields

```
public static final System.Drawing.SizeF Empty;
```

SolidBrush Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SolidBrush
   extends System.Drawing.Brush
```

Public Constructors

```
public SolidBrush(
   System.Drawing.Color color
);
```

Public Properties

```
public System.Drawing.Color get_Color();
public void set_Color(
   System.Drawing.Color value
);
```

Public Methods

```
public System.Object Clone();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

StringAlignment Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum StringAlignment
```

Members

Center

Far

Near

StringDigitSubstitute Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum StringDigitSubstitute
```

Members

National

None

Traditional

User

StringFormat Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class StringFormat
   extends System.MarshalByRefObject
   implements System.ICloneable, System.IDisposable
```

Public Constructors

```
public StringFormat();
public StringFormat(
   System.Drawing.StringFormat format
);
public StringFormat(
   System.Drawing.StringFormatFlags options
);
public StringFormat(
   System.Drawing.StringFormatFlags options,
   int language
);
```

Public Properties

```
public System.Drawing.StringAlignment get_Alignment();
public void set_Alignment(
   System.Drawing.StringAlignment value
);
public int get_DigitSubstitutionLanguage();
public System.Drawing.StringDigitSubstitute get_DigitSubstitutionMethod();
public System.Drawing.StringFormatFlags get_FormatFlags();
public void set_FormatFlags(
   System.Drawing.StringFormatFlags value
);
public static System.Drawing.StringFormat get_GenericDefault();
public static System.Drawing.StringFormat get_GenericTypographic();
public System.Drawing.Text.HotkeyPrefix get_HotkeyPrefix();
public void set_HotkeyPrefix(
   System.Drawing.Text.HotkeyPrefix value
);
public System.Drawing.StringAlignment get_LineAlignment();
public void set_LineAlignment(
   System.Drawing.StringAlignment value
);
public System.Drawing.StringTrimming get_Trimming();
public void set_Trimming(
   System.Drawing.StringTrimming value
);
```

Public Methods

```
public final System.Object Clone();
public final void Dispose();
public float[] GetTabStops(
   float firstTabOffset
);
public void SetDigitSubstitution(
```

```
      int language,
      System.Drawing.StringDigitSubstitute substitute
   );
   public void SetMeasurableCharacterRanges(
      System.Drawing.CharacterRange[] ranges
   );
   public void SetTabStops(
      float firstTabOffset,
      float[] tabStops
   );
   public System.String ToString();
```

Protected Methods

```
   protected void Finalize();
```

StringFormatFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum StringFormatFlags
```

Members

DirectionRightToLeft

DirectionVertical

DisplayFormatControl

FitBlackBox

LineLimit

MeasureTrailingSpaces

NoClip

NoFontFallback

NoWrap

StringTrimming Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum StringTrimming
```

Members

Character

EllipsisCharacter

EllipsisPath

EllipsisWord

None

Word

StringUnit Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum StringUnit
```

Members

Display

Document

Em

Inch

Millimeter

Pixel

Point

World

SystemBrushes Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SystemBrushes
   extends System.Object
```

Public Properties

```
public static System.Drawing.Brush get_ActiveBorder();
public static System.Drawing.Brush get_ActiveCaption();
public static System.Drawing.Brush get_ActiveCaptionText();
public static System.Drawing.Brush get_AppWorkspace();
public static System.Drawing.Brush get_Control();
public static System.Drawing.Brush get_ControlDark();
public static System.Drawing.Brush get_ControlDarkDark();
public static System.Drawing.Brush get_ControlLight();
public static System.Drawing.Brush get_ControlLightLight();
public static System.Drawing.Brush get_ControlText();
public static System.Drawing.Brush get_Desktop();
public static System.Drawing.Brush get_Highlight();
public static System.Drawing.Brush get_HighlightText();
public static System.Drawing.Brush get_HotTrack();
public static System.Drawing.Brush get_InactiveBorder();
public static System.Drawing.Brush get_InactiveCaption();
public static System.Drawing.Brush get_Info();
public static System.Drawing.Brush get_Menu();
public static System.Drawing.Brush get_ScrollBar();
public static System.Drawing.Brush get_Window();
public static System.Drawing.Brush get_WindowText();
```

Public Methods

```
public static System.Drawing.Brush FromSystemColor(
   System.Drawing.Color c
);
```

SystemColors Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SystemColors
   extends System.Object
```

Public Properties

```
public static System.Drawing.Color get_ActiveBorder();
public static System.Drawing.Color get_ActiveCaption();
public static System.Drawing.Color get_ActiveCaptionText();
public static System.Drawing.Color get_AppWorkspace();
public static System.Drawing.Color get_Control();
public static System.Drawing.Color get_ControlDark();
public static System.Drawing.Color get_ControlDarkDark();
public static System.Drawing.Color get_ControlLight();
public static System.Drawing.Color get_ControlLightLight();
public static System.Drawing.Color get_ControlText();
public static System.Drawing.Color get_Desktop();
public static System.Drawing.Color get_GrayText();
public static System.Drawing.Color get_Highlight();
public static System.Drawing.Color get_HighlightText();
public static System.Drawing.Color get_HotTrack();
public static System.Drawing.Color get_InactiveBorder();
public static System.Drawing.Color get_InactiveCaption();
public static System.Drawing.Color get_InactiveCaptionText();
public static System.Drawing.Color get_Info();
public static System.Drawing.Color get_InfoText();
public static System.Drawing.Color get_Menu();
public static System.Drawing.Color get_MenuText();
public static System.Drawing.Color get_ScrollBar();
public static System.Drawing.Color get_Window();
public static System.Drawing.Color get_WindowFrame();
public static System.Drawing.Color get_WindowText();
```

SystemIcons Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SystemIcons
   extends System.Object
```

Public Properties

```
public static System.Drawing.Icon get_Application();
public static System.Drawing.Icon get_Asterisk();
public static System.Drawing.Icon get_Error();
public static System.Drawing.Icon get_Exclamation();
public static System.Drawing.Icon get_Hand();
public static System.Drawing.Icon get_Information();
public static System.Drawing.Icon get_Question();
public static System.Drawing.Icon get_Warning();
public static System.Drawing.Icon get_WinLogo();
```

SystemPens Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SystemPens
   extends System.Object
```

Public Properties

```
public static System.Drawing.Pen get_ActiveCaptionText();
public static System.Drawing.Pen get_Control();
public static System.Drawing.Pen get_ControlDark();
public static System.Drawing.Pen get_ControlDarkDark();
public static System.Drawing.Pen get_ControlLight();
public static System.Drawing.Pen get_ControlLightLight();
public static System.Drawing.Pen get_ControlText();
public static System.Drawing.Pen get_GrayText();
public static System.Drawing.Pen get_Highlight();
public static System.Drawing.Pen get_HighlightText();
public static System.Drawing.Pen get_InactiveCaptionText();
public static System.Drawing.Pen get_InfoText();
public static System.Drawing.Pen get_MenuText();
public static System.Drawing.Pen get_WindowFrame();
public static System.Drawing.Pen get_WindowText();
```

Public Methods

```
public static System.Drawing.Pen FromSystemColor(
   System.Drawing.Color c
);
```

TextureBrush Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TextureBrush
   extends System.Drawing.Brush
```

Public Constructors

```
public TextureBrush(
   System.Drawing.Image bitmap
);
public TextureBrush(
   System.Drawing.Image image,
   System.Drawing.Drawing2D.WrapMode wrapMode
);
public TextureBrush(
   System.Drawing.Image image,
   System.Drawing.Rectangle dstRect
);
public TextureBrush(
   System.Drawing.Image image,
   System.Drawing.RectangleF dstRect
);
public TextureBrush(
   System.Drawing.Image image,
   System.Drawing.Drawing2D.WrapMode wrapMode,
   System.Drawing.Rectangle dstRect
);
public TextureBrush(
   System.Drawing.Image image,
   System.Drawing.Drawing2D.WrapMode wrapMode,
   System.Drawing.RectangleF dstRect
);
public TextureBrush(
   System.Drawing.Image image,
   System.Drawing.Rectangle dstRect,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
public TextureBrush(
   System.Drawing.Image image,
   System.Drawing.RectangleF dstRect,
   System.Drawing.Imaging.ImageAttributes imageAttr
);
```

Public Properties

```
public System.Drawing.Image get_Image();
public System.Drawing.Drawing2D.Matrix get_Transform();
public void set_Transform(
   System.Drawing.Drawing2D.Matrix value
);
public System.Drawing.Drawing2D.WrapMode get_WrapMode();
public void set_WrapMode(
   System.Drawing.Drawing2D.WrapMode value
);
```

Public Methods

```
public System.Object Clone();
public void MultiplyTransform(
   System.Drawing.Drawing2D.Matrix matrix
);
public void MultiplyTransform(
   System.Drawing.Drawing2D.Matrix matrix,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void ResetTransform();
public void RotateTransform(
   float angle
);
public void RotateTransform(
   float angle,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void ScaleTransform(
   float sx,
   float sy
);
public void ScaleTransform(
   float sx,
   float sy,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void TranslateTransform(
   float dx,
   float dy
);
public void TranslateTransform(
   float dx,
   float dy,
   System.Drawing.Drawing2D.MatrixOrder order
);
```

ToolboxBitmapAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class ToolboxBitmapAttribute
   extends System.Attribute
```

Public Constructors

```
public ToolboxBitmapAttribute(
   System.String imageFile
);
public ToolboxBitmapAttribute(
   System.Type t
);
public ToolboxBitmapAttribute(
   System.Type t,
   System.String name
);
```

Public Methods

```
public boolean Equals(
   System.Object value
);
public int GetHashCode();
public System.Drawing.Image GetImage(
   System.Object component
);
public System.Drawing.Image GetImage(
   System.Type type
);
public System.Drawing.Image GetImage(
   System.Object component,
   boolean large
);
public System.Drawing.Image GetImage(
   System.Type type,
   boolean large
);
public System.Drawing.Image GetImage(
   System.Type type,
   System.String imgName,
   boolean large
);
public static System.Drawing.Image GetImageFromResource(
   System.Type t,
   System.String imageName,
   boolean large
);
```

Public Fields

```
public static final System.Drawing.ToolboxBitmapAttribute Default;
```

System.Drawing.Design Namespace (Visual J# Syntax)

Classes

BitmapEditor

CategoryNameCollection

ColorEditor

ContentAlignmentEditor

CursorEditor

FontEditor

FontNameEditor

IconEditor

ImageEditor

MetafileEditor

PaintValueEventArgs

PropertyValueUIItem

ToolboxComponentsCreatedEventArgs

ToolboxComponentsCreatingEventArgs

ToolboxItem

ToolboxItemCollection

UITypeEditor

Interfaces

IPropertyValueUIService

IToolboxService

IToolboxUser

Delegates

PropertyValueUIHandler

PropertyValueUIItemInvokeHandler

ToolboxComponentsCreatedEventHandler

ToolboxComponentsCreatingEventHandler

ToolboxItemCreatorCallback

Enumerations

UITypeEditorEditStyle

BitmapEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BitmapEditor
   extends System.Drawing.Design.ImageEditor
```

Public Constructors

```
public BitmapEditor();
```

Protected Methods

```
protected System.String[] GetExtensions();
protected System.String GetFileDialogDescription();
protected System.Drawing.Image LoadFromStream(
   System.IO.Stream stream
);
```

CategoryNameCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CategoryNameCollection
   extends System.Collections.ReadOnlyCollectionBase
```

Public Constructors

```
public CategoryNameCollection(
   System.Drawing.Design.CategoryNameCollection value
);
public CategoryNameCollection(
   System.String[] value
);
```

Public Properties

```
public System.String get_Item(
   int index
);
```

Public Methods

```
public boolean Contains(
   System.String value
);
public void CopyTo(
   System.String[] array,
   int index
);
public int IndexOf(
   System.String value
);
```

ColorEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ColorEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public ColorEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetPaintValueSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public void PaintValue(
   System.Drawing.Design.PaintValueEventArgs e
);
```

ContentAlignmentEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ContentAlignmentEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public ContentAlignmentEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

CursorEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CursorEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public CursorEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

FontEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FontEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public FontEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

FontNameEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FontNameEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public FontNameEditor();
```

Public Methods

```
public boolean GetPaintValueSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public void PaintValue(
   System.Drawing.Design.PaintValueEventArgs e
);
```

IconEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class IconEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public IconEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetPaintValueSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public void PaintValue(
   System.Drawing.Design.PaintValueEventArgs e
);
```

Protected Methods

```
protected static System.String CreateExtensionsString(
   System.String[] extensions,
   System.String sep
);
protected static System.String CreateFilterEntry(
   System.Drawing.Design.IconEditor e
);
protected System.String[] GetExtensions();
protected System.String GetFileDialogDescription();
protected System.Drawing.Icon LoadFromStream(
   System.IO.Stream stream
);
```

ImageEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ImageEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public ImageEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetPaintValueSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public void PaintValue(
   System.Drawing.Design.PaintValueEventArgs e
);
```

Protected Methods

```
protected static System.String CreateExtensionsString(
   System.String[] extensions,
   System.String sep
);
protected static System.String CreateFilterEntry(
   System.Drawing.Design.ImageEditor e
);
protected System.String[] GetExtensions();
protected System.String GetFileDialogDescription();
protected System.Drawing.Image LoadFromStream(
   System.IO.Stream stream
);
```

IPropertyValueUIService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IPropertyValueUIService
```

Public Methods

```
void AddPropertyValueUIHandler(
   System.Drawing.Design.PropertyValueUIHandler newHandler
);
System.Drawing.Design.PropertyValueUIItem[] GetPropertyUIValueItems(
   System.ComponentModel.ITypeDescriptorContext context,
   System.ComponentModel.PropertyDescriptor propDesc
);
void NotifyPropertyValueUIItemsChanged();
void RemovePropertyValueUIHandler(
   System.Drawing.Design.PropertyValueUIHandler newHandler
);
```

Public Events

```
void add_PropertyUIValueItemsChanged(
   System.EventHandler value
);
void remove_PropertyUIValueItemsChanged(
   System.EventHandler value
);
```

IToolboxService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "4BACD258-DE64-4048-BC4E-
FEDBEF9ACB76") */
public interface IToolboxService
```

Public Properties

```
System.Drawing.Design.CategoryNameCollection get_CategoryNames();
System.String get_SelectedCategory();
void set_SelectedCategory(
   System.String value
);
```

Public Methods

```
void AddCreator(
   System.Drawing.Design.ToolboxItemCreatorCallback creator,
   System.String format
);
void AddCreator(
   System.Drawing.Design.ToolboxItemCreatorCallback creator,
   System.String format,
   System.ComponentModel.Design.IDesignerHost host
);
void AddLinkedToolboxItem(
   System.Drawing.Design.ToolboxItem toolboxItem,
   System.ComponentModel.Design.IDesignerHost host
);
void AddLinkedToolboxItem(
   System.Drawing.Design.ToolboxItem toolboxItem,
   System.String category,
   System.ComponentModel.Design.IDesignerHost host
);
void AddToolboxItem(
   System.Drawing.Design.ToolboxItem toolboxItem
);
void AddToolboxItem(
   System.Drawing.Design.ToolboxItem toolboxItem,
   System.String category
);
System.Drawing.Design.ToolboxItem DeserializeToolboxItem(
   System.Object serializedObject
);
System.Drawing.Design.ToolboxItem DeserializeToolboxItem(
   System.Object serializedObject,
   System.ComponentModel.Design.IDesignerHost host
);
System.Drawing.Design.ToolboxItem GetSelectedToolboxItem();
System.Drawing.Design.ToolboxItem GetSelectedToolboxItem(
   System.ComponentModel.Design.IDesignerHost host
);
System.Drawing.Design.ToolboxItemCollection GetToolboxItems();
System.Drawing.Design.ToolboxItemCollection GetToolboxItems(
   System.ComponentModel.Design.IDesignerHost host
```

```
);
System.Drawing.Design.ToolboxItemCollection GetToolboxItems(
   System.String category
);
System.Drawing.Design.ToolboxItemCollection GetToolboxItems(
   System.String category,
   System.ComponentModel.Design.IDesignerHost host
);
boolean IsSupported(
   System.Object serializedObject,
   System.Collections.ICollection filterAttributes
);
boolean IsSupported(
   System.Object serializedObject,
   System.ComponentModel.Design.IDesignerHost host
);
boolean IsToolboxItem(
   System.Object serializedObject
);
boolean IsToolboxItem(
   System.Object serializedObject,
   System.ComponentModel.Design.IDesignerHost host
);
void Refresh();
void RemoveCreator(
   System.String format
);
void RemoveCreator(
   System.String format,
   System.ComponentModel.Design.IDesignerHost host
);
void RemoveToolboxItem(
   System.Drawing.Design.ToolboxItem toolboxItem
);
void RemoveToolboxItem(
   System.Drawing.Design.ToolboxItem toolboxItem,
   System.String category
);
void SelectedToolboxItemUsed();
System.Object SerializeToolboxItem(
   System.Drawing.Design.ToolboxItem toolboxItem
);
boolean SetCursor();
void SetSelectedToolboxItem(
   System.Drawing.Design.ToolboxItem toolboxItem
);
```

IToolboxUser Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IToolboxUser
```

Public Methods

```
boolean GetToolSupported(
   System.Drawing.Design.ToolboxItem tool
);
void ToolPicked(
   System.Drawing.Design.ToolboxItem tool
);
```

MetafileEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MetafileEditor
   extends System.Drawing.Design.ImageEditor
```

Public Constructors

```
public MetafileEditor();
```

Protected Methods

```
protected System.String[] GetExtensions();
protected System.String GetFileDialogDescription();
protected System.Drawing.Image LoadFromStream(
   System.IO.Stream stream
);
```

PaintValueEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PaintValueEventArgs
   extends System.EventArgs
```

Public Constructors

```
public PaintValueEventArgs(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value,
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle bounds
);
```

Public Properties

```
public System.Drawing.Rectangle get_Bounds();
public System.ComponentModel.ITypeDescriptorContext get_Context();
public System.Drawing.Graphics get_Graphics();
public System.Object get_Value();
```

PropertyValueUIHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void PropertyValueUIHandler(
   System.ComponentModel.ITypeDescriptorContext context,
   System.ComponentModel.PropertyDescriptor propDesc,
   System.Collections.ArrayList valueUIItemList
);
```

PropertyValueUIItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PropertyValueUIItem
   extends System.Object
```

Public Constructors

```
public PropertyValueUIItem(
   System.Drawing.Image uiItemImage,
   System.Drawing.Design.PropertyValueUIItemInvokeHandler handler,
   System.String tooltip
);
```

Public Properties

```
public System.Drawing.Image get_Image();
public System.Drawing.Design.PropertyValueUIItemInvokeHandler get_InvokeHandler();
public System.String get_ToolTip();
```

Public Methods

```
public void Reset();
```

PropertyValueUIItemInvokeHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void PropertyValueUIItemInvokeHandler(
   System.ComponentModel.ITypeDescriptorContext context,
   System.ComponentModel.PropertyDescriptor descriptor,
   System.Drawing.Design.PropertyValueUIItem invokedItem
);
```

ToolboxComponentsCreatedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ToolboxComponentsCreatedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ToolboxComponentsCreatedEventArgs(
   System.ComponentModel.IComponent[] components
);
```

Public Properties

```
public System.ComponentModel.IComponent[] get_Components();
```

ToolboxComponentsCreatedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ToolboxComponentsCreatedEventHandler(
   System.Object sender,
   System.Drawing.Design.ToolboxComponentsCreatedEventArgs e
);
```

ToolboxComponentsCreatingEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ToolboxComponentsCreatingEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ToolboxComponentsCreatingEventArgs(
   System.ComponentModel.Design.IDesignerHost host
);
```

Public Properties

```
public System.ComponentModel.Design.IDesignerHost get_DesignerHost();
```

ToolboxComponentsCreatingEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ToolboxComponentsCreatingEventHandler(
   System.Object sender,
   System.Drawing.Design.ToolboxComponentsCreatingEventArgs e
);
```

ToolboxItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ToolboxItem
   extends System.Object
   implements System.Runtime.Serialization.ISerializable
```

Public Constructors

```
public ToolboxItem();
public ToolboxItem(
   System.Type toolType
);
```

Public Properties

```
public System.Reflection.AssemblyName get_AssemblyName();
public void set_AssemblyName(
   System.Reflection.AssemblyName value
);
public System.Drawing.Bitmap get_Bitmap();
public void set_Bitmap(
   System.Drawing.Bitmap value
);
public System.String get_DisplayName();
public void set_DisplayName(
   System.String value
);
public System.Collections.ICollection get_Filter();
public void set_Filter(
   System.Collections.ICollection value
);
public System.String get_TypeName();
public void set_TypeName(
   System.String value
);
```

Public Methods

```
public System.ComponentModel.IComponent[] CreateComponents();
public System.ComponentModel.IComponent[] CreateComponents(
   System.ComponentModel.Design.IDesignerHost host
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public void Initialize(
   System.Type type
);
public void Lock();
public System.String ToString();
```

Public Events

```
public void add_ComponentsCreated(
   System.Drawing.Design.ToolboxComponentsCreatedEventHandler value
);
public void remove_ComponentsCreated(
   System.Drawing.Design.ToolboxComponentsCreatedEventHandler value
);
public void add_ComponentsCreating(
   System.Drawing.Design.ToolboxComponentsCreatingEventHandler value
);
public void remove_ComponentsCreating(
   System.Drawing.Design.ToolboxComponentsCreatingEventHandler value
);
```

Protected Properties

```
protected boolean get_Locked();
```

Protected Methods

```
protected void CheckUnlocked();
protected System.ComponentModel.IComponent[] CreateComponentsCore(
   System.ComponentModel.Design.IDesignerHost host
);
protected void Deserialize(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
protected System.Type GetType(
   System.ComponentModel.Design.IDesignerHost host,
   System.Reflection.AssemblyName assemblyName,
   System.String typeName,
   boolean reference
);
protected void OnComponentsCreated(
   System.Drawing.Design.ToolboxComponentsCreatedEventArgs args
);
protected void OnComponentsCreating(
   System.Drawing.Design.ToolboxComponentsCreatingEventArgs args
);
protected void Serialize(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ToolboxItemCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ToolboxItemCollection
   extends System.Collections.ReadOnlyCollectionBase
```

Public Constructors

```
public ToolboxItemCollection(
   System.Drawing.Design.ToolboxItemCollection value
);
public ToolboxItemCollection(
   System.Drawing.Design.ToolboxItem[] value
);
```

Public Properties

```
public System.Drawing.Design.ToolboxItem get_Item(
   int index
);
```

Public Methods

```
public boolean Contains(
   System.Drawing.Design.ToolboxItem value
);
public void CopyTo(
   System.Drawing.Design.ToolboxItem[] array,
   int index
);
public int IndexOf(
   System.Drawing.Design.ToolboxItem value
);
```

ToolboxItemCreatorCallback Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
System.Drawing.Design.ToolboxItem ToolboxItemCreatorCallback(
   System.Object serializedObject,
   System.String format
);
```

UITypeEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UITypeEditor
   extends System.Object
```

Public Constructors

```
public UITypeEditor();
```

Public Methods

```
public System.Object EditValue(
   System.IServiceProvider provider,
   System.Object value
);
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle();
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetPaintValueSupported();
public boolean GetPaintValueSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public void PaintValue(
   System.Drawing.Design.PaintValueEventArgs e
);
public void PaintValue(
   System.Object value,
   System.Drawing.Graphics canvas,
   System.Drawing.Rectangle rectangle
);
```

UITypeEditorEditStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum UITypeEditorEditStyle
```

Members

DropDown

Modal

None

System.Drawing.Drawing2D Namespace (Visual J# Syntax)

Classes

AdjustableArrowCap

Blend

ColorBlend

CustomLineCap

GraphicsContainer

GraphicsPath

GraphicsPathIterator

GraphicsState

HatchBrush

LinearGradientBrush

Matrix

PathData

PathGradientBrush

RegionData

Enumerations

CombineMode

CompositingMode

CompositingQuality

CoordinateSpace

DashCap

DashStyle

FillMode

FlushIntention

HatchStyle

InterpolationMode

LinearGradientMode

LineCap

LineJoin

MatrixOrder

PathPointType

PenAlignment

PenType

PixelOffsetMode

QualityMode

SmoothingMode

WarpMode

WrapMode

AdjustableArrowCap Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AdjustableArrowCap
   extends System.Drawing.Drawing2D.CustomLineCap
```

Public Constructors

```
public AdjustableArrowCap(
   float width,
   float height
);
public AdjustableArrowCap(
   float width,
   float height,
   boolean isFilled
);
```

Public Properties

```
public boolean get_Filled();
public void set_Filled(
   boolean value
);
public float get_Height();
public void set_Height(
   float value
);
public float get_MiddleInset();
public void set_MiddleInset(
   float value
);
public float get_Width();
public void set_Width(
   float value
);
```

Blend Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Blend
   extends System.Object
```

Public Constructors

```
public Blend();
public Blend(
   int count
);
```

Public Properties

```
public float[] get_Factors();
public void set_Factors(
   float[] value
);
public float[] get_Positions();
public void set_Positions(
   float[] value
);
```

ColorBlend Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ColorBlend
   extends System.Object
```

Public Constructors

```
public ColorBlend();
public ColorBlend(
   int count
);
```

Public Properties

```
public System.Drawing.Color[] get_Colors();
public void set_Colors(
   System.Drawing.Color[] value
);
public float[] get_Positions();
public void set_Positions(
   float[] value
);
```

CombineMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CombineMode
```

Members

Complement

Exclude

Intersect

Replace

Union

Xor

CompositingMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CompositingMode
```

Members

SourceCopy

SourceOver

CompositingQuality Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CompositingQuality
```

Members

AssumeLinear

Default

GammaCorrected

HighQuality

HighSpeed

Invalid

CoordinateSpace Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CoordinateSpace
```

Members

Device

Page

World

CustomLineCap Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CustomLineCap
   extends System.MarshalByRefObject
   implements System.ICloneable, System.IDisposable
```

Public Constructors

```
public CustomLineCap(
   System.Drawing.Drawing2D.GraphicsPath fillPath,
   System.Drawing.Drawing2D.GraphicsPath strokePath
);
public CustomLineCap(
   System.Drawing.Drawing2D.GraphicsPath fillPath,
   System.Drawing.Drawing2D.GraphicsPath strokePath,
   System.Drawing.Drawing2D.LineCap baseCap
);
public CustomLineCap(
   System.Drawing.Drawing2D.GraphicsPath fillPath,
   System.Drawing.Drawing2D.GraphicsPath strokePath,
   System.Drawing.Drawing2D.LineCap baseCap,
   float baseInset
);
```

Public Properties

```
public System.Drawing.Drawing2D.LineCap get_BaseCap();
public void set_BaseCap(
   System.Drawing.Drawing2D.LineCap value
);
public float get_BaseInset();
public void set_BaseInset(
   float value
);
public System.Drawing.Drawing2D.LineJoin get_StrokeJoin();
public void set_StrokeJoin(
   System.Drawing.Drawing2D.LineJoin value
);
public float get_WidthScale();
public void set_WidthScale(
   float value
);
```

Public Methods

```
public final System.Object Clone();
public final void Dispose();
public void GetStrokeCaps(
   System.Drawing.Drawing2D.LineCap& startCap,
   System.Drawing.Drawing2D.LineCap& endCap
);
public void SetStrokeCaps(
   System.Drawing.Drawing2D.LineCap startCap,
   System.Drawing.Drawing2D.LineCap endCap
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

DashCap Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum DashCap
```

Members

Flat

Round

Triangle

DashStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum DashStyle
```

Members

Custom

Dash

DashDot

DashDotDot

Dot

Solid

FillMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum FillMode
```

Members

Alternate

Winding

FlushIntention Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum FlushIntention
```

Members

Flush

Sync

GraphicsContainer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class GraphicsContainer
   extends System.MarshalByRefObject
```

GraphicsPath Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class GraphicsPath
   extends System.MarshalByRefObject
   implements System.ICloneable, System.IDisposable
```

Public Constructors

```
public GraphicsPath();
public GraphicsPath(
   System.Drawing.Drawing2D.FillMode fillMode
);
public GraphicsPath(
   System.Drawing.PointF[] pts,
   ubyte[] types
);
public GraphicsPath(
   System.Drawing.Point[] pts,
   ubyte[] types
);
public GraphicsPath(
   System.Drawing.PointF[] pts,
   ubyte[] types,
   System.Drawing.Drawing2D.FillMode fillMode
);
public GraphicsPath(
   System.Drawing.Point[] pts,
   ubyte[] types,
   System.Drawing.Drawing2D.FillMode fillMode
);
```

Public Properties

```
public System.Drawing.Drawing2D.FillMode get_FillMode();
public void set_FillMode(
   System.Drawing.Drawing2D.FillMode value
);
public System.Drawing.Drawing2D.PathData get_PathData();
public System.Drawing.PointF[] get_PathPoints();
public ubyte[] get_PathTypes();
public int get_PointCount();
```

Public Methods

```
public void AddArc(
   System.Drawing.Rectangle rect,
   float startAngle,
   float sweepAngle
);
public void AddArc(
   System.Drawing.RectangleF rect,
   float startAngle,
   float sweepAngle
);
public void AddArc(
```

```
    int x,
    int y,
    int width,
    int height,
    float startAngle,
    float sweepAngle
);
public void AddArc(
    float x,
    float y,
    float width,
    float height,
    float startAngle,
    float sweepAngle
);
public void AddBezier(
    System.Drawing.Point pt1,
    System.Drawing.Point pt2,
    System.Drawing.Point pt3,
    System.Drawing.Point pt4
);
public void AddBezier(
    System.Drawing.PointF pt1,
    System.Drawing.PointF pt2,
    System.Drawing.PointF pt3,
    System.Drawing.PointF pt4
);
public void AddBezier(
    int x1,
    int y1,
    int x2,
    int y2,
    int x3,
    int y3,
    int x4,
    int y4
);
public void AddBezier(
    float x1,
    float y1,
    float x2,
    float y2,
    float x3,
    float y3,
    float x4,
    float y4
);
public void AddBeziers(
    System.Drawing.PointF[] points
);
public void AddBeziers(
    System.Drawing.Point[] points
);
public void AddClosedCurve(
    System.Drawing.PointF[] points
);
public void AddClosedCurve(
    System.Drawing.Point[] points
);
public void AddClosedCurve(
    System.Drawing.PointF[] points,
    float tension
);
public void AddClosedCurve(
    System.Drawing.Point[] points,
    float tension
);
public void AddCurve(
```

```
   System.Drawing.PointF[] points
);
public void AddCurve(
   System.Drawing.Point[] points
);
public void AddCurve(
   System.Drawing.PointF[] points,
   float tension
);
public void AddCurve(
   System.Drawing.Point[] points,
   float tension
);
public void AddCurve(
   System.Drawing.PointF[] points,
   int offset,
   int numberOfSegments,
   float tension
);
public void AddCurve(
   System.Drawing.Point[] points,
   int offset,
   int numberOfSegments,
   float tension
);
public void AddEllipse(
   System.Drawing.Rectangle rect
);
public void AddEllipse(
   System.Drawing.RectangleF rect
);
public void AddEllipse(
   int x,
   int y,
   int width,
   int height
);
public void AddEllipse(
   float x,
   float y,
   float width,
   float height
);
public void AddLine(
   System.Drawing.Point pt1,
   System.Drawing.Point pt2
);
public void AddLine(
   System.Drawing.PointF pt1,
   System.Drawing.PointF pt2
);
public void AddLine(
   int x1,
   int y1,
   int x2,
   int y2
);
public void AddLine(
   float x1,
   float y1,
   float x2,
   float y2
);
public void AddLines(
   System.Drawing.PointF[] points
);
public void AddLines(
   System.Drawing.Point[] points
```

```
);
public void AddPath(
   System.Drawing.Drawing2D.GraphicsPath addingPath,
   boolean connect
);
public void AddPie(
   System.Drawing.Rectangle rect,
   float startAngle,
   float sweepAngle
);
public void AddPie(
   int x,
   int y,
   int width,
   int height,
   float startAngle,
   float sweepAngle
);
public void AddPie(
   float x,
   float y,
   float width,
   float height,
   float startAngle,
   float sweepAngle
);
public void AddPolygon(
   System.Drawing.PointF[] points
);
public void AddPolygon(
   System.Drawing.Point[] points
);
public void AddRectangle(
   System.Drawing.Rectangle rect
);
public void AddRectangle(
   System.Drawing.RectangleF rect
);
public void AddRectangles(
   System.Drawing.RectangleF[] rects
);
public void AddRectangles(
   System.Drawing.Rectangle[] rects
);
public void AddString(
   System.String s,
   System.Drawing.FontFamily family,
   int style,
   float emSize,
   System.Drawing.Point origin,
   System.Drawing.StringFormat format
);
public void AddString(
   System.String s,
   System.Drawing.FontFamily family,
   int style,
   float emSize,
   System.Drawing.PointF origin,
   System.Drawing.StringFormat format
);
public void AddString(
   System.String s,
   System.Drawing.FontFamily family,
   int style,
   float emSize,
   System.Drawing.Rectangle layoutRect,
   System.Drawing.StringFormat format
);
```

```
public void AddString(
   System.String s,
   System.Drawing.FontFamily family,
   int style,
   float emSize,
   System.Drawing.RectangleF layoutRect,
   System.Drawing.StringFormat format
);
public void ClearMarkers();
public final System.Object Clone();
public void CloseAllFigures();
public void CloseFigure();
public final void Dispose();
public void Flatten();
public void Flatten(
   System.Drawing.Drawing2D.Matrix matrix
);
public void Flatten(
   System.Drawing.Drawing2D.Matrix matrix,
   float flatness
);
public System.Drawing.RectangleF GetBounds();
public System.Drawing.RectangleF GetBounds(
   System.Drawing.Drawing2D.Matrix matrix
);
public System.Drawing.RectangleF GetBounds(
   System.Drawing.Drawing2D.Matrix matrix,
   System.Drawing.Pen pen
);
public System.Drawing.PointF GetLastPoint();
public boolean IsOutlineVisible(
   System.Drawing.Point point,
   System.Drawing.Pen pen
);
public boolean IsOutlineVisible(
   System.Drawing.PointF point,
   System.Drawing.Pen pen
);
public boolean IsOutlineVisible(
   int x,
   int y,
   System.Drawing.Pen pen
);
public boolean IsOutlineVisible(
   float x,
   float y,
   System.Drawing.Pen pen
);
public boolean IsOutlineVisible(
   System.Drawing.Point pt,
   System.Drawing.Pen pen,
   System.Drawing.Graphics graphics
);
public boolean IsOutlineVisible(
   System.Drawing.PointF pt,
   System.Drawing.Pen pen,
   System.Drawing.Graphics graphics
);
public boolean IsOutlineVisible(
   int x,
   int y,
   System.Drawing.Pen pen,
   System.Drawing.Graphics graphics
);
public boolean IsOutlineVisible(
   float x,
   float y,
   System.Drawing.Pen pen,
```

```
   System.Drawing.Graphics graphics
);
public boolean IsVisible(
   System.Drawing.Point point
);
public boolean IsVisible(
   System.Drawing.PointF point
);
public boolean IsVisible(
   int x,
   int y
);
public boolean IsVisible(
   float x,
   float y
);
public boolean IsVisible(
   System.Drawing.Point pt,
   System.Drawing.Graphics graphics
);
public boolean IsVisible(
   System.Drawing.PointF pt,
   System.Drawing.Graphics graphics
);
public boolean IsVisible(
   int x,
   int y,
   System.Drawing.Graphics graphics
);
public boolean IsVisible(
   float x,
   float y,
   System.Drawing.Graphics graphics
);
public void Reset();
public void Reverse();
public void SetMarkers();
public void StartFigure();
public void Transform(
   System.Drawing.Drawing2D.Matrix matrix
);
public void Warp(
   System.Drawing.PointF[] destPoints,
   System.Drawing.RectangleF srcRect
);
public void Warp(
   System.Drawing.PointF[] destPoints,
   System.Drawing.RectangleF srcRect,
   System.Drawing.Drawing2D.Matrix matrix
);
public void Warp(
   System.Drawing.PointF[] destPoints,
   System.Drawing.RectangleF srcRect,
   System.Drawing.Drawing2D.Matrix matrix,
   System.Drawing.Drawing2D.WarpMode warpMode
);
public void Warp(
   System.Drawing.PointF[] destPoints,
   System.Drawing.RectangleF srcRect,
   System.Drawing.Drawing2D.Matrix matrix,
   System.Drawing.Drawing2D.WarpMode warpMode,
   float flatness
);
public void Widen(
   System.Drawing.Pen pen
);
public void Widen(
   System.Drawing.Pen pen,
```

```
   System.Drawing.Drawing2D.Matrix matrix
);
public void Widen(
   System.Drawing.Pen pen,
   System.Drawing.Drawing2D.Matrix matrix,
   float flatness
);
```

Protected Methods

```
protected void Finalize();
```

GraphicsPathIterator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class GraphicsPathIterator
   extends System.MarshalByRefObject
   implements System.IDisposable
```

Public Constructors

```
public GraphicsPathIterator(
   System.Drawing.Drawing2D.GraphicsPath path
);
```

Public Properties

```
public int get_Count();
public int get_SubpathCount();
```

Public Methods

```
public int CopyData(
   System.Drawing.PointF[]& points,
   System.Byte[]& types,
   int startIndex,
   int endIndex
);
public final void Dispose();
public int Enumerate(
   System.Drawing.PointF[]& points,
   System.Byte[]& types
);
public boolean HasCurve();
public int NextMarker(
   System.Drawing.Drawing2D.GraphicsPath path
);
public int NextMarker(
   int startIndex,
   int endIndex
);
public int NextPathType(
   ubyte pathType,
   int startIndex,
   int endIndex
);
public int NextSubpath(
   System.Drawing.Drawing2D.GraphicsPath path,
   boolean isClosed
);
public int NextSubpath(
   int startIndex,
   int endIndex,
   boolean isClosed
);
public void Rewind();
```

Protected Methods

```
protected void Finalize();
```

GraphicsState Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class GraphicsState
   extends System.MarshalByRefObject
```

HatchBrush Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HatchBrush
   extends System.Drawing.Brush
```

Public Constructors

```
public HatchBrush(
   System.Drawing.Drawing2D.HatchStyle hatchstyle,
   System.Drawing.Color foreColor
);
public HatchBrush(
   System.Drawing.Drawing2D.HatchStyle hatchstyle,
   System.Drawing.Color foreColor,
   System.Drawing.Color backColor
);
```

Public Properties

```
public System.Drawing.Color get_BackgroundColor();
public System.Drawing.Color get_ForegroundColor();
public System.Drawing.Drawing2D.HatchStyle get_HatchStyle();
```

Public Methods

```
public System.Object Clone();
```

HatchStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum HatchStyle
```

Members

BackwardDiagonal

Cross

DarkDownwardDiagonal

DarkHorizontal

DarkUpwardDiagonal

DarkVertical

DashedDownwardDiagonal

DashedHorizontal

DashedUpwardDiagonal

DashedVertical

DiagonalBrick

DiagonalCross

Divot

DottedDiamond

DottedGrid

ForwardDiagonal

Horizontal

HorizontalBrick

LargeCheckerBoard

LargeConfetti

LargeGrid

LightDownwardDiagonal

LightHorizontal

LightUpwardDiagonal

LightVertical

Max

Min

NarrowHorizontal

NarrowVertical

OutlinedDiamond

Percent05

Percent10

Percent20

Percent25

Percent30

Percent40

Percent50

Percent60

Percent70

Percent75

Percent80

Percent90

Plaid

Shingle

SmallCheckerBoard

SmallConfetti

SmallGrid

SolidDiamond

Sphere

Trellis

Vertical

Wave

Weave

WideDownwardDiagonal

WideUpwardDiagonal

ZigZag

InterpolationMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum InterpolationMode
```

Members

Bicubic

Bilinear

Default

High

HighQualityBicubic

HighQualityBilinear

Invalid

Low

NearestNeighbor

LinearGradientBrush Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LinearGradientBrush
   extends System.Drawing.Brush
```

Public Constructors

```
public LinearGradientBrush(
   System.Drawing.Point point1,
   System.Drawing.Point point2,
   System.Drawing.Color color1,
   System.Drawing.Color color2
);
public LinearGradientBrush(
   System.Drawing.PointF point1,
   System.Drawing.PointF point2,
   System.Drawing.Color color1,
   System.Drawing.Color color2
);
public LinearGradientBrush(
   System.Drawing.Rectangle rect,
   System.Drawing.Color color1,
   System.Drawing.Color color2,
   float angle
);
public LinearGradientBrush(
   System.Drawing.Rectangle rect,
   System.Drawing.Color color1,
   System.Drawing.Color color2,
   System.Drawing.Drawing2D.LinearGradientMode linearGradientMode
);
public LinearGradientBrush(
   System.Drawing.RectangleF rect,
   System.Drawing.Color color1,
   System.Drawing.Color color2,
   float angle
);
public LinearGradientBrush(
   System.Drawing.RectangleF rect,
   System.Drawing.Color color1,
   System.Drawing.Color color2,
   System.Drawing.Drawing2D.LinearGradientMode linearGradientMode
);
public LinearGradientBrush(
   System.Drawing.Rectangle rect,
   System.Drawing.Color color1,
   System.Drawing.Color color2,
   float angle,
   boolean isAngleScaleable
);
public LinearGradientBrush(
   System.Drawing.RectangleF rect,
   System.Drawing.Color color1,
   System.Drawing.Color color2,
   float angle,
   boolean isAngleScaleable
);
```

Public Properties

```
public System.Drawing.Drawing2D.Blend get_Blend();
public void set_Blend(
   System.Drawing.Drawing2D.Blend value
);
public boolean get_GammaCorrection();
public void set_GammaCorrection(
   boolean value
);
public System.Drawing.Drawing2D.ColorBlend get_InterpolationColors();
public void set_InterpolationColors(
   System.Drawing.Drawing2D.ColorBlend value
);
public System.Drawing.Color[] get_LinearColors();
public void set_LinearColors(
   System.Drawing.Color[] value
);
public System.Drawing.RectangleF get_Rectangle();
public System.Drawing.Drawing2D.Matrix get_Transform();
public void set_Transform(
   System.Drawing.Drawing2D.Matrix value
);
public System.Drawing.Drawing2D.WrapMode get_WrapMode();
public void set_WrapMode(
   System.Drawing.Drawing2D.WrapMode value
);
```

Public Methods

```
public System.Object Clone();
public void MultiplyTransform(
   System.Drawing.Drawing2D.Matrix matrix
);
public void MultiplyTransform(
   System.Drawing.Drawing2D.Matrix matrix,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void ResetTransform();
public void RotateTransform(
   float angle
);
public void RotateTransform(
   float angle,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void ScaleTransform(
   float sx,
   float sy
);
public void ScaleTransform(
   float sx,
   float sy,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void SetBlendTriangularShape(
   float focus
);
public void SetBlendTriangularShape(
   float focus,
   float scale
);
public void SetSigmaBellShape(
   float focus
);
```

```
public void SetSigmaBellShape(
   float focus,
   float scale
);
public void TranslateTransform(
   float dx,
   float dy
);
public void TranslateTransform(
   float dx,
   float dy,
   System.Drawing.Drawing2D.MatrixOrder order
);
```

LinearGradientMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum LinearGradientMode
```

Members

BackwardDiagonal

ForwardDiagonal

Horizontal

Vertical

LineCap Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum LineCap
```

Members

AnchorMask

ArrowAnchor

Custom

DiamondAnchor

Flat

NoAnchor

Round

RoundAnchor

Square

SquareAnchor

Triangle

LineJoin Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum LineJoin
```

Members

Bevel

Miter

MiterClipped

Round

Matrix Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Matrix
   extends System.MarshalByRefObject
   implements System.IDisposable
```

Public Constructors

```
public Matrix();
public Matrix(
   System.Drawing.Rectangle rect,
   System.Drawing.Point[] plgpts
);
public Matrix(
   System.Drawing.RectangleF rect,
   System.Drawing.PointF[] plgpts
);
public Matrix(
   float m11,
   float m12,
   float m21,
   float m22,
   float dx,
   float dy
);
```

Public Properties

```
public float[] get_Elements();
public boolean get_IsIdentity();
public boolean get_IsInvertible();
public float get_OffsetX();
public float get_OffsetY();
```

Public Methods

```
public System.Drawing.Drawing2D.Matrix Clone();
public final void Dispose();
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public void Invert();
public void Multiply(
   System.Drawing.Drawing2D.Matrix matrix
);
public void Multiply(
   System.Drawing.Drawing2D.Matrix matrix,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void Reset();
public void Rotate(
   float angle
);
public void Rotate(
```

```
   float angle,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void RotateAt(
   float angle,
   System.Drawing.PointF point
);
public void RotateAt(
   float angle,
   System.Drawing.PointF point,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void Scale(
   float scaleX,
   float scaleY
);
public void Scale(
   float scaleX,
   float scaleY,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void Shear(
   float shearX,
   float shearY
);
public void Shear(
   float shearX,
   float shearY,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void TransformPoints(
   System.Drawing.PointF[] pts
);
public void TransformPoints(
   System.Drawing.Point[] pts
);
public void TransformVectors(
   System.Drawing.PointF[] pts
);
public void TransformVectors(
   System.Drawing.Point[] pts
);
public void Translate(
   float offsetX,
   float offsetY
);
public void Translate(
   float offsetX,
   float offsetY,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void VectorTransformPoints(
   System.Drawing.Point[] pts
);
```

Protected Methods

```
protected void Finalize();
```

MatrixOrder Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MatrixOrder
```

Members

Append

Prepend

PathData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PathData
   extends System.Object
```

Public Constructors

```
public PathData();
```

Public Properties

```
public System.Drawing.PointF[] get_Points();
public void set_Points(
   System.Drawing.PointF[] value
);
public ubyte[] get_Types();
public void set_Types(
   ubyte[] value
);
```

PathGradientBrush Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PathGradientBrush
   extends System.Drawing.Brush
```

Public Constructors

```
public PathGradientBrush(
   System.Drawing.Drawing2D.GraphicsPath path
);
public PathGradientBrush(
   System.Drawing.PointF[] points
);
public PathGradientBrush(
   System.Drawing.Point[] points
);
public PathGradientBrush(
   System.Drawing.PointF[] points,
   System.Drawing.Drawing2D.WrapMode wrapMode
);
public PathGradientBrush(
   System.Drawing.Point[] points,
   System.Drawing.Drawing2D.WrapMode wrapMode
);
```

Public Properties

```
public System.Drawing.Drawing2D.Blend get_Blend();
public void set_Blend(
   System.Drawing.Drawing2D.Blend value
);
public System.Drawing.Color get_CenterColor();
public void set_CenterColor(
   System.Drawing.Color value
);
public System.Drawing.PointF get_CenterPoint();
public void set_CenterPoint(
   System.Drawing.PointF value
);
public System.Drawing.PointF get_FocusScales();
public void set_FocusScales(
   System.Drawing.PointF value
);
public System.Drawing.Drawing2D.ColorBlend get_InterpolationColors();
public void set_InterpolationColors(
   System.Drawing.Drawing2D.ColorBlend value
);
public System.Drawing.RectangleF get_Rectangle();
public System.Drawing.Color[] get_SurroundColors();
public void set_SurroundColors(
   System.Drawing.Color[] value
);
public System.Drawing.Drawing2D.Matrix get_Transform();
public void set_Transform(
   System.Drawing.Drawing2D.Matrix value
);
public System.Drawing.Drawing2D.WrapMode get_WrapMode();
```

```
public void set_WrapMode(
   System.Drawing.Drawing2D.WrapMode value
);
```

Public Methods

```
public System.Object Clone();
public void MultiplyTransform(
   System.Drawing.Drawing2D.Matrix matrix
);
public void MultiplyTransform(
   System.Drawing.Drawing2D.Matrix matrix,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void ResetTransform();
public void RotateTransform(
   float angle
);
public void RotateTransform(
   float angle,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void ScaleTransform(
   float sx,
   float sy
);
public void ScaleTransform(
   float sx,
   float sy,
   System.Drawing.Drawing2D.MatrixOrder order
);
public void SetBlendTriangularShape(
   float focus
);
public void SetBlendTriangularShape(
   float focus,
   float scale
);
public void SetSigmaBellShape(
   float focus
);
public void SetSigmaBellShape(
   float focus,
   float scale
);
public void TranslateTransform(
   float dx,
   float dy
);
public void TranslateTransform(
   float dx,
   float dy,
   System.Drawing.Drawing2D.MatrixOrder order
);
```

PathPointType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PathPointType
```

Members

Bezier

Bezier3

CloseSubpath

DashMode

Line

PathMarker

PathTypeMask

Start

PenAlignment Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PenAlignment
```

Members

Center

Inset

Left

Outset

Right

PenType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PenType
```

Members

HatchFill

LinearGradient

PathGradient

SolidColor

TextureFill

PixelOffsetMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PixelOffsetMode
```

Members

Default

Half

HighQuality

HighSpeed

Invalid

None

QualityMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum QualityMode
```

Members

Default

High

Invalid

Low

RegionData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RegionData
   extends System.Object
```

Public Properties

```
public ubyte[] get_Data();
public void set_Data(
   ubyte[] value
);
```

SmoothingMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SmoothingMode
```

Members

AntiAlias

Default

HighQuality

HighSpeed

Invalid

None

WarpMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum WarpMode
```

Members

Bilinear

Perspective

WrapMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum WrapMode
```

Members

Clamp

Tile

TileFlipX

TileFlipXY

TileFlipY

System.Drawing.Imaging Namespace (Visual J# Syntax)

Classes

BitmapData

ColorMap

ColorMatrix

ColorPalette

Encoder

EncoderParameter

EncoderParameters

FrameDimension

ImageAttributes

ImageCodecInfo

ImageFormat

Metafile

MetafileHeader

MetaHeader

PropertyItem

WmfPlaceableFileHeader

Delegates

PlayRecordCallback

Enumerations

ColorAdjustType

ColorChannelFlag

ColorMapType

ColorMatrixFlag

ColorMode

EmfPlusRecordType

EmfType

EncoderParameterValueType

EncoderValue

ImageCodecFlags

ImageFlags

ImageLockMode

MetafileFrameUnit

MetafileType

PaletteFlags

PixelFormat

BitmapData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BitmapData
   extends System.Object
```

Public Constructors

```
public BitmapData();
```

Public Properties

```
public int get_Height();
public void set_Height(
   int value
);
public System.Drawing.Imaging.PixelFormat get_PixelFormat();
public void set_PixelFormat(
   System.Drawing.Imaging.PixelFormat value
);
public int get_Reserved();
public void set_Reserved(
   int value
);
public System.IntPtr get_Scan0();
public void set_Scan0(
   System.IntPtr value
);
public int get_Stride();
public void set_Stride(
   int value
);
public int get_Width();
public void set_Width(
   int value
);
```

ColorAdjustType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ColorAdjustType
```

Members

Any

Bitmap

Brush

Count

Default

Pen

Text

ColorChannelFlag Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ColorChannelFlag
```

Members

ColorChannelC

ColorChannelK

ColorChannelLast

ColorChannelM

ColorChannelY

ColorMap Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ColorMap
   extends System.Object
```

Public Constructors

```
public ColorMap();
```

Public Properties

```
public System.Drawing.Color get_NewColor();
public void set_NewColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_OldColor();
public void set_OldColor(
   System.Drawing.Color value
);
```

ColorMapType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ColorMapType
```

Members

Brush

Default

ColorMatrix Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ColorMatrix
   extends System.Object
```

Public Constructors

```
public ColorMatrix();
```

Public Properties

```
public float get_Item(
   int row,
   int column
);
public void set_Item(
   int row,
   int column,
   float value
);
public float get_Matrix00();
public void set_Matrix00(
   float value
);
public float get_Matrix01();
public void set_Matrix01(
   float value
);
public float get_Matrix02();
public void set_Matrix02(
   float value
);
public float get_Matrix03();
public void set_Matrix03(
   float value
);
public float get_Matrix04();
public void set_Matrix04(
   float value
);
public float get_Matrix10();
public void set_Matrix10(
   float value
);
public float get_Matrix11();
public void set_Matrix11(
   float value
);
public float get_Matrix12();
public void set_Matrix12(
   float value
);
public float get_Matrix13();
public void set_Matrix13(
   float value
);
```

```
public float get_Matrix14();
public void set_Matrix14(
   float value
);
public float get_Matrix20();
public void set_Matrix20(
   float value
);
public float get_Matrix21();
public void set_Matrix21(
   float value
);
public float get_Matrix22();
public void set_Matrix22(
   float value
);
public float get_Matrix23();
public void set_Matrix23(
   float value
);
public float get_Matrix24();
public void set_Matrix24(
   float value
);
public float get_Matrix30();
public void set_Matrix30(
   float value
);
public float get_Matrix31();
public void set_Matrix31(
   float value
);
public float get_Matrix32();
public void set_Matrix32(
   float value
);
public float get_Matrix33();
public void set_Matrix33(
   float value
);
public float get_Matrix34();
public void set_Matrix34(
   float value
);
public float get_Matrix40();
public void set_Matrix40(
   float value
);
public float get_Matrix41();
public void set_Matrix41(
   float value
);
public float get_Matrix42();
public void set_Matrix42(
   float value
);
public float get_Matrix43();
public void set_Matrix43(
   float value
);
public float get_Matrix44();
public void set_Matrix44(
   float value
);
```

ColorMatrixFlag Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ColorMatrixFlag
```

Members

AltGrays

Default

SkipGrays

ColorMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ColorMode
```

Members

Argb32Mode

Argb64Mode

ColorPalette Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ColorPalette
   extends System.Object
```

Public Properties

```
public System.Drawing.Color[] get_Entries();
public int get_Flags();
```

EmfPlusRecordType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum EmfPlusRecordType
```

Members

BeginContainer

BeginContainerNoParams

Clear

Comment

DrawArc

DrawBeziers

DrawClosedCurve

DrawCurve

DrawDriverString

DrawEllipse

DrawImage

DrawImagePoints

DrawLines

DrawPath

DrawPie

DrawRects

DrawString

EmfAbortPath

EmfAlphaBlend

EmfAngleArc

EmfArcTo

EmfBeginPath

EmfBitBlt

EmfChord

EmfCloseFigure

EmfColorCorrectPalette

EmfColorMatchToTargetW

EmfCreateBrushIndirect

EmfCreateColorSpace

EmfCreateColorSpaceW

EmfCreateDibPatternBrushPt

EmfCreateMonoBrush

EmfCreatePalette

EmfCreatePen

EmfDeleteColorSpace

EmfDeleteObject

EmfDrawEscape

EmfEllipse

EmfEndPath

EmfEof

EmfExcludeClipRect

EmfExtCreateFontIndirect

EmfExtCreatePen

EmfExtEscape

EmfExtFloodFill

EmfExtSelectClipRgn

EmfExtTextOutA

EmfExtTextOutW

EmfFillPath

EmfFillRgn

EmfFlattenPath

EmfForceUfiMapping

EmfFrameRgn

EmfGdiComment

EmfGlsBoundedRecord

EmfGlsRecord

EmfGradientFill

EmfHeader

EmfIntersectClipRect

EmfInvertRgn

EmfLineTo

EmfMaskBlt

EmfMax

EmfMin

EmfModifyWorldTransform

EmfMoveToEx

EmfNamedEscpae

EmfOffsetClipRgn

EmfPaintRgn

EmfPie

EmfPixelFormat

EmfPlgBlt

EmfPlusRecordBase

EmfPolyBezier

EmfPolyBezier16

EmfPolyBezierTo

EmfPolyBezierTo16

EmfPolyDraw

EmfPolyDraw16

EmfPolygon

EmfPolygon16

EmfPolyline

EmfPolyline16

EmfPolyLineTo

EmfPolylineTo16

EmfPolyPolygon

EmfPolyPolygon16

EmfPolyPolyline

EmfPolyPolyline16

EmfPolyTextOutA

EmfPolyTextOutW

EmfRealizePalette

EmfRectangle

EmfReserved069

EmfReserved117

EmfResizePalette

EmfRestoreDC

EmfRoundArc

EmfRoundRect

EmfSaveDC

EmfScaleViewportExtEx

EmfScaleWindowExtEx

EmfSelectClipPath

EmfSelectObject

EmfSelectPalette

EmfSetArcDirection

EmfSetBkColor

EmfSetBkMode

EmfSetBrushOrgEx

EmfSetColorAdjustment

EmfSetColorSpace

EmfSetDIBitsToDevice

EmfSetIcmMode

EmfSetIcmProfileA

EmfSetIcmProfileW

EmfSetLayout

EmfSetLinkedUfis

EmfSetMapMode

EmfSetMapperFlags

EmfSetMetaRgn

EmfSetMiterLimit

EmfSetPaletteEntries

EmfSetPixelV

EmfSetPolyFillMode

EmfSetROP2

EmfSetStretchBltMode

EmfSetTextAlign

EmfSetTextColor

EmfSetTextJustification

EmfSetViewportExtEx

EmfSetViewportOrgEx

EmfSetWindowExtEx

EmfSetWindowOrgEx

EmfSetWorldTransform

EmfSmallTextOut

EmfStartDoc

EmfStretchBlt

EmfStretchDIBits

EmfStrokeAndFillPath

EmfStrokePath

EmfTransparentBlt

EmfWidenPath

EndContainer

EndOfFile

FillClosedCurve

FillEllipse

FillPath

FillPie

FillPolygon

FillRects

FillRegion

GetDC

Header

Invalid

Max

Min

MultiFormatEnd

MultiFormatSection

MultiFormatStart

MultiplyWorldTransform

Object

OffsetClip

ResetClip

ResetWorldTransform

Restore

RotateWorldTransform

Save

ScaleWorldTransform

SetAntiAliasMode

SetClipPath

SetClipRect

SetClipRegion

SetCompositingMode

SetCompositingQuality

SetInterpolationMode

SetPageTransform

SetPixelOffsetMode

SetRenderingOrigin

SetTextContrast

SetTextRenderingHint

SetWorldTransform

Total

TranslateWorldTransform

WmfAnimatePalette

WmfArc

WmfBitBlt

WmfChord

WmfCreateBrushIndirect

WmfCreateFontIndirect

WmfCreatePalette

WmfCreatePatternBrush

WmfCreatePenIndirect

WmfCreateRegion

WmfDeleteObject

WmfDibBitBlt

WmfDibCreatePatternBrush

WmfDibStretchBlt

WmfEllipse

WmfEscape

WmfExcludeClipRect

WmfExtFloodFill

WmfExtTextOut

WmfFillRegion

WmfFloodFill

WmfFrameRegion

WmfIntersectClipRect

WmfInvertRegion

WmfLineTo

WmfMoveTo

WmfOffsetCilpRgn

WmfOffsetViewportOrg

WmfOffsetWindowOrg

WmfPaintRegion

WmfPatBlt

WmfPie

WmfPolygon

WmfPolyline

WmfPolyPolygon

WmfRealizePalette

WmfRecordBase

WmfRectangle

WmfResizePalette

WmfRestoreDC

WmfRoundRect

WmfSaveDC

WmfScaleViewportExt

WmfScaleWindowExt

WmfSelectClipRegion

WmfSelectObject

WmfSelectPalette

WmfSetBkColor

WmfSetBkMode

WmfSetDibToDev

WmfSetLayout

WmfSetMapMode

WmfSetMapperFlags

WmfSetPalEntries

WmfSetPixel

WmfSetPolyFillMode

WmfSetRelAbs

WmfSetROP2

WmfSetStretchBltMode

WmfSetTextAlign

WmfSetTextCharExtra

WmfSetTextColor

WmfSetTextJustification

WmfSetViewportExt

WmfSetViewportOrg

WmfSetWindowExt

WmfSetWindowOrg

WmfStretchBlt

WmfStretchDib

WmfTextOut

EmfType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum EmfType
```

Members

EmfOnly

EmfPlusDual

EmfPlusOnly

Encoder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Encoder
   extends System.Object
```

Public Constructors

```
public Encoder(
   System.Guid guid
);
```

Public Properties

```
public System.Guid get_Guid();
```

Public Fields

```
public static final System.Drawing.Imaging.Encoder ChrominanceTable;
public static final System.Drawing.Imaging.Encoder ColorDepth;
public static final System.Drawing.Imaging.Encoder Compression;
public static final System.Drawing.Imaging.Encoder LuminanceTable;
public static final System.Drawing.Imaging.Encoder Quality;
public static final System.Drawing.Imaging.Encoder RenderMethod;
public static final System.Drawing.Imaging.Encoder SaveFlag;
public static final System.Drawing.Imaging.Encoder ScanMethod;
public static final System.Drawing.Imaging.Encoder Transformation;
public static final System.Drawing.Imaging.Encoder Version;
```

EncoderParameter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EncoderParameter
   extends System.Object
   implements System.IDisposable
```

Public Constructors

```
public EncoderParameter(
   System.Drawing.Imaging.Encoder encoder,
   ubyte value
);
public EncoderParameter(
   System.Drawing.Imaging.Encoder encoder,
   ubyte[] value
);
public EncoderParameter(
   System.Drawing.Imaging.Encoder encoder,
   short value
);
public EncoderParameter(
   System.Drawing.Imaging.Encoder encoder,
   short[] value
);
public EncoderParameter(
   System.Drawing.Imaging.Encoder encoder,
   long value
);
public EncoderParameter(
   System.Drawing.Imaging.Encoder encoder,
   long[] value
);
public EncoderParameter(
   System.Drawing.Imaging.Encoder encoder,
   System.String value
);
public EncoderParameter(
   System.Drawing.Imaging.Encoder encoder,
   ubyte value,
   boolean undefined
);
public EncoderParameter(
   System.Drawing.Imaging.Encoder encoder,
   ubyte[] value,
   boolean undefined
);
public EncoderParameter(
   System.Drawing.Imaging.Encoder encoder,
   int numerator,
   int demoninator
);
public EncoderParameter(
   System.Drawing.Imaging.Encoder encoder,
   int[] numerator,
   int[] denominator
);
public EncoderParameter(
   System.Drawing.Imaging.Encoder encoder,
   long rangebegin,
```

```
      long rangeend
   );
   public EncoderParameter(
      System.Drawing.Imaging.Encoder encoder,
      long[] rangebegin,
      long[] rangeend
   );
   public EncoderParameter(
      System.Drawing.Imaging.Encoder encoder,
      int NumberOfValues,
      int Type,
      int Value
   );
   public EncoderParameter(
      System.Drawing.Imaging.Encoder encoder,
      int numerator1,
      int demoninator1,
      int numerator2,
      int demoninator2
   );
   public EncoderParameter(
      System.Drawing.Imaging.Encoder encoder,
      int[] numerator1,
      int[] denominator1,
      int[] numerator2,
      int[] denominator2
   );
```

Public Properties

```
   public System.Drawing.Imaging.Encoder get_Encoder();
   public void set_Encoder(
      System.Drawing.Imaging.Encoder value
   );
   public int get_NumberOfValues();
   public System.Drawing.Imaging.EncoderParameterValueType get_Type();
   public System.Drawing.Imaging.EncoderParameterValueType get_ValueType();
```

Public Methods

```
   public final void Dispose();
```

Protected Methods

```
   protected void Finalize();
```

EncoderParameters Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EncoderParameters
   extends System.Object
   implements System.IDisposable
```

Public Constructors

```
public EncoderParameters();
public EncoderParameters(
   int count
);
```

Public Properties

```
public System.Drawing.Imaging.EncoderParameter[] get_Param();
public void set_Param(
   System.Drawing.Imaging.EncoderParameter[] value
);
```

Public Methods

```
public final void Dispose();
```

EncoderParameterValueType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum EncoderParameterValueType
```

Members

ValueTypeAscii

ValueTypeByte

ValueTypeLong

ValueTypeLongRange

ValueTypeRational

ValueTypeRationalRange

ValueTypeShort

ValueTypeUndefined

EncoderValue Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum EncoderValue
```

Members

ColorTypeCMYK

ColorTypeYCCK

CompressionCCITT3

CompressionCCITT4

CompressionLZW

CompressionNone

CompressionRle

Flush

FrameDimensionPage

FrameDimensionResolution

FrameDimensionTime

LastFrame

MultiFrame

RenderNonProgressive

RenderProgressive

ScanMethodInterlaced

ScanMethodNonInterlaced

TransformFlipHorizontal

TransformFlipVertical

TransformRotate180

TransformRotate270

TransformRotate90

VersionGif87

VersionGif89

FrameDimension Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FrameDimension
   extends System.Object
```

Public Constructors

```
public FrameDimension(
   System.Guid guid
);
```

Public Properties

```
public System.Guid get_Guid();
public static System.Drawing.Imaging.FrameDimension get_Page();
public static System.Drawing.Imaging.FrameDimension get_Resolution();
public static System.Drawing.Imaging.FrameDimension get_Time();
```

Public Methods

```
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public System.String ToString();
```

ImageAttributes Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ImageAttributes
   extends System.Object
   implements System.ICloneable, System.IDisposable
```

Public Constructors

```
public ImageAttributes();
```

Public Methods

```
public void ClearBrushRemapTable();
public void ClearColorKey();
public void ClearColorKey(
   System.Drawing.Imaging.ColorAdjustType type
);
public void ClearColorMatrix();
public void ClearColorMatrix(
   System.Drawing.Imaging.ColorAdjustType type
);
public void ClearGamma();
public void ClearGamma(
   System.Drawing.Imaging.ColorAdjustType type
);
public void ClearNoOp();
public void ClearNoOp(
   System.Drawing.Imaging.ColorAdjustType type
);
public void ClearOutputChannel();
public void ClearOutputChannel(
   System.Drawing.Imaging.ColorAdjustType type
);
public void ClearOutputChannelColorProfile();
public void ClearOutputChannelColorProfile(
   System.Drawing.Imaging.ColorAdjustType type
);
public void ClearRemapTable();
public void ClearRemapTable(
   System.Drawing.Imaging.ColorAdjustType type
);
public void ClearThreshold();
public void ClearThreshold(
   System.Drawing.Imaging.ColorAdjustType type
);
public final System.Object Clone();
public final void Dispose();
public void GetAdjustedPalette(
   System.Drawing.Imaging.ColorPalette palette,
   System.Drawing.Imaging.ColorAdjustType type
);
public void SetBrushRemapTable(
   System.Drawing.Imaging.ColorMap[] map
);
public void SetColorKey(
   System.Drawing.Color colorLow,
   System.Drawing.Color colorHigh
```

```
);
public void SetColorKey(
   System.Drawing.Color colorLow,
   System.Drawing.Color colorHigh,
   System.Drawing.Imaging.ColorAdjustType type
);
public void SetColorMatrices(
   System.Drawing.Imaging.ColorMatrix newColorMatrix,
   System.Drawing.Imaging.ColorMatrix grayMatrix
);
public void SetColorMatrices(
   System.Drawing.Imaging.ColorMatrix newColorMatrix,
   System.Drawing.Imaging.ColorMatrix grayMatrix,
   System.Drawing.Imaging.ColorMatrixFlag flags
);
public void SetColorMatrices(
   System.Drawing.Imaging.ColorMatrix newColorMatrix,
   System.Drawing.Imaging.ColorMatrix grayMatrix,
   System.Drawing.Imaging.ColorMatrixFlag mode,
   System.Drawing.Imaging.ColorAdjustType type
);
public void SetColorMatrix(
   System.Drawing.Imaging.ColorMatrix newColorMatrix
);
public void SetColorMatrix(
   System.Drawing.Imaging.ColorMatrix newColorMatrix,
   System.Drawing.Imaging.ColorMatrixFlag flags
);
public void SetColorMatrix(
   System.Drawing.Imaging.ColorMatrix newColorMatrix,
   System.Drawing.Imaging.ColorMatrixFlag mode,
   System.Drawing.Imaging.ColorAdjustType type
);
public void SetGamma(
   float gamma
);
public void SetGamma(
   float gamma,
   System.Drawing.Imaging.ColorAdjustType type
);
public void SetNoOp();
public void SetNoOp(
   System.Drawing.Imaging.ColorAdjustType type
);
public void SetOutputChannel(
   System.Drawing.Imaging.ColorChannelFlag flags
);
public void SetOutputChannel(
   System.Drawing.Imaging.ColorChannelFlag flags,
   System.Drawing.Imaging.ColorAdjustType type
);
public void SetOutputChannelColorProfile(
   System.String colorProfileFilename
);
public void SetOutputChannelColorProfile(
   System.String colorProfileFilename,
   System.Drawing.Imaging.ColorAdjustType type
);
public void SetRemapTable(
   System.Drawing.Imaging.ColorMap[] map
);
public void SetRemapTable(
   System.Drawing.Imaging.ColorMap[] map,
   System.Drawing.Imaging.ColorAdjustType type
);
public void SetThreshold(
   float threshold
);
```

```
public void SetThreshold(
   float threshold,
   System.Drawing.Imaging.ColorAdjustType type
);
public void SetWrapMode(
   System.Drawing.Drawing2D.WrapMode mode
);
public void SetWrapMode(
   System.Drawing.Drawing2D.WrapMode mode,
   System.Drawing.Color color
);
public void SetWrapMode(
   System.Drawing.Drawing2D.WrapMode mode,
   System.Drawing.Color color,
   boolean clamp
);
```

Protected Methods

```
protected void Finalize();
```

ImageCodecFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum ImageCodecFlags
```

Members

BlockingDecode

Builtin

Decoder

Encoder

SeekableEncode

SupportBitmap

SupportVector

System

User

ImageCodecInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class ImageCodecInfo
   extends System.Object
```

Public Properties

```
public System.Guid get_Clsid();
public void set_Clsid(
   System.Guid value
);
public System.String get_CodecName();
public void set_CodecName(
   System.String value
);
public System.String get_DllName();
public void set_DllName(
   System.String value
);
public System.String get_FilenameExtension();
public void set_FilenameExtension(
   System.String value
);
public System.Drawing.Imaging.ImageCodecFlags get_Flags();
public void set_Flags(
   System.Drawing.Imaging.ImageCodecFlags value
);
public System.String get_FormatDescription();
public void set_FormatDescription(
   System.String value
);
public System.Guid get_FormatID();
public void set_FormatID(
   System.Guid value
);
public System.String get_MimeType();
public void set_MimeType(
   System.String value
);
public System.Byte[][] get_SignatureMasks();
public void set_SignatureMasks(
   System.Byte[][] value
);
public System.Byte[][] get_SignaturePatterns();
public void set_SignaturePatterns(
   System.Byte[][] value
);
public int get_Version();
public void set_Version(
   int value
);
```

Public Methods

```
public static System.Drawing.Imaging.ImageCodecInfo[] GetImageDecoders();
public static System.Drawing.Imaging.ImageCodecInfo[] GetImageEncoders();
```

ImageFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum ImageFlags
```

Members

Caching

ColorSpaceCmyk

ColorSpaceGray

ColorSpaceRgb

ColorSpaceYcbcr

ColorSpaceYcck

HasAlpha

HasRealDpi

HasRealPixelSize

HasTranslucent

None

PartiallyScalable

ReadOnly

Scalable

ImageFormat Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Drawi
ng.ImageFormatConverter, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=
b03f5f7f11d50a3a") */
public class ImageFormat
   extends System.Object
```

Public Constructors

```
public ImageFormat(
   System.Guid guid
);
```

Public Properties

```
public static System.Drawing.Imaging.ImageFormat get_Bmp();
public static System.Drawing.Imaging.ImageFormat get_Emf();
public static System.Drawing.Imaging.ImageFormat get_Exif();
public static System.Drawing.Imaging.ImageFormat get_Gif();
public System.Guid get_Guid();
public static System.Drawing.Imaging.ImageFormat get_Icon();
public static System.Drawing.Imaging.ImageFormat get_Jpeg();
public static System.Drawing.Imaging.ImageFormat get_MemoryBmp();
public static System.Drawing.Imaging.ImageFormat get_Png();
public static System.Drawing.Imaging.ImageFormat get_Tiff();
public static System.Drawing.Imaging.ImageFormat get_Wmf();
```

Public Methods

```
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public System.String ToString();
```

ImageLockMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ImageLockMode
```

Members

ReadOnly

ReadWrite

UserInputBuffer

WriteOnly

Metafile Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Drawing.Design.MetafileEditor, System.Drawing.Design, Ve
rsion=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class Metafile
   extends System.Drawing.Image
```

Public Constructors

```
public Metafile(
   System.IO.Stream stream
);
public Metafile(
   System.String filename
);
public Metafile(
   System.IntPtr henhmetafile,
   boolean deleteEmf
);
public Metafile(
   System.IntPtr referenceHdc,
   System.Drawing.Imaging.EmfType emfType
);
public Metafile(
   System.IntPtr hmetafile,
   System.Drawing.Imaging.WmfPlaceableFileHeader wmfHeader
);
public Metafile(
   System.IntPtr referenceHdc,
   System.Drawing.Rectangle frameRect
);
public Metafile(
   System.IntPtr referenceHdc,
   System.Drawing.RectangleF frameRect
);
public Metafile(
   System.IO.Stream stream,
   System.IntPtr referenceHdc
);
public Metafile(
   System.String fileName,
   System.IntPtr referenceHdc
);
public Metafile(
   System.IntPtr referenceHdc,
   System.Drawing.Imaging.EmfType emfType,
   System.String description
);
public Metafile(
   System.IntPtr hmetafile,
   System.Drawing.Imaging.WmfPlaceableFileHeader wmfHeader,
   boolean deleteWmf
);
public Metafile(
   System.IntPtr referenceHdc,
```

```
   System.Drawing.Rectangle frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit
);
public Metafile(
   System.IntPtr referenceHdc,
   System.Drawing.RectangleF frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit
);
public Metafile(
   System.IO.Stream stream,
   System.IntPtr referenceHdc,
   System.Drawing.Imaging.EmfType type
);
public Metafile(
   System.IO.Stream stream,
   System.IntPtr referenceHdc,
   System.Drawing.Rectangle frameRect
);
public Metafile(
   System.IO.Stream stream,
   System.IntPtr referenceHdc,
   System.Drawing.RectangleF frameRect
);
public Metafile(
   System.String fileName,
   System.IntPtr referenceHdc,
   System.Drawing.Imaging.EmfType type
);
public Metafile(
   System.String fileName,
   System.IntPtr referenceHdc,
   System.Drawing.Rectangle frameRect
);
public Metafile(
   System.String fileName,
   System.IntPtr referenceHdc,
   System.Drawing.RectangleF frameRect
);
public Metafile(
   System.IntPtr referenceHdc,
   System.Drawing.Rectangle frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit,
   System.Drawing.Imaging.EmfType type
);
public Metafile(
   System.IntPtr referenceHdc,
   System.Drawing.RectangleF frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit,
   System.Drawing.Imaging.EmfType type
);
public Metafile(
   System.IO.Stream stream,
   System.IntPtr referenceHdc,
   System.Drawing.Imaging.EmfType type,
   System.String description
);
public Metafile(
   System.IO.Stream stream,
   System.IntPtr referenceHdc,
   System.Drawing.Rectangle frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit
);
public Metafile(
   System.IO.Stream stream,
   System.IntPtr referenceHdc,
   System.Drawing.RectangleF frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit
);
```

```
public Metafile(
   System.String fileName,
   System.IntPtr referenceHdc,
   System.Drawing.Imaging.EmfType type,
   System.String description
);
public Metafile(
   System.String fileName,
   System.IntPtr referenceHdc,
   System.Drawing.Rectangle frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit
);
public Metafile(
   System.String fileName,
   System.IntPtr referenceHdc,
   System.Drawing.RectangleF frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit
);
public Metafile(
   System.IntPtr referenceHdc,
   System.Drawing.Rectangle frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit,
   System.Drawing.Imaging.EmfType type,
   System.String desc
);
public Metafile(
   System.IntPtr referenceHdc,
   System.Drawing.RectangleF frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit,
   System.Drawing.Imaging.EmfType type,
   System.String description
);
public Metafile(
   System.IO.Stream stream,
   System.IntPtr referenceHdc,
   System.Drawing.Rectangle frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit,
   System.Drawing.Imaging.EmfType type
);
public Metafile(
   System.IO.Stream stream,
   System.IntPtr referenceHdc,
   System.Drawing.RectangleF frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit,
   System.Drawing.Imaging.EmfType type
);
public Metafile(
   System.String fileName,
   System.IntPtr referenceHdc,
   System.Drawing.Rectangle frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit,
   System.Drawing.Imaging.EmfType type
);
public Metafile(
   System.String fileName,
   System.IntPtr referenceHdc,
   System.Drawing.Rectangle frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit,
   System.String description
);
public Metafile(
   System.String fileName,
   System.IntPtr referenceHdc,
   System.Drawing.RectangleF frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit,
   System.Drawing.Imaging.EmfType type
);
public Metafile(
```

```
   System.String fileName,
   System.IntPtr referenceHdc,
   System.Drawing.RectangleF frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit,
   System.String desc
);
public Metafile(
   System.IO.Stream stream,
   System.IntPtr referenceHdc,
   System.Drawing.Rectangle frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit,
   System.Drawing.Imaging.EmfType type,
   System.String description
);
public Metafile(
   System.IO.Stream stream,
   System.IntPtr referenceHdc,
   System.Drawing.RectangleF frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit,
   System.Drawing.Imaging.EmfType type,
   System.String description
);
public Metafile(
   System.String fileName,
   System.IntPtr referenceHdc,
   System.Drawing.Rectangle frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit,
   System.Drawing.Imaging.EmfType type,
   System.String description
);
public Metafile(
   System.String fileName,
   System.IntPtr referenceHdc,
   System.Drawing.RectangleF frameRect,
   System.Drawing.Imaging.MetafileFrameUnit frameUnit,
   System.Drawing.Imaging.EmfType type,
   System.String description
);
```

Public Methods

```
public System.IntPtr GetHenhmetafile();
public System.Drawing.Imaging.MetafileHeader GetMetafileHeader();
public static System.Drawing.Imaging.MetafileHeader GetMetafileHeader(
   System.IntPtr henhmetafile
);
public static System.Drawing.Imaging.MetafileHeader GetMetafileHeader(
   System.IO.Stream stream
);
public static System.Drawing.Imaging.MetafileHeader GetMetafileHeader(
   System.String fileName
);
public static System.Drawing.Imaging.MetafileHeader GetMetafileHeader(
   System.IntPtr hmetafile,
   System.Drawing.Imaging.WmfPlaceableFileHeader wmfHeader
);
public void PlayRecord(
   System.Drawing.Imaging.EmfPlusRecordType recordType,
   int flags,
   int dataSize,
   ubyte[] data
);
```

MetafileFrameUnit Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MetafileFrameUnit
```

Members

Document

GdiCompatible

Inch

Millimeter

Pixel

Point

MetafileHeader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MetafileHeader
   extends System.Object
```

Public Properties

```
public System.Drawing.Rectangle get_Bounds();
public float get_DpiX();
public float get_DpiY();
public int get_EmfPlusHeaderSize();
public int get_LogicalDpiX();
public int get_LogicalDpiY();
public int get_MetafileSize();
public System.Drawing.Imaging.MetafileType get_Type();
public int get_Version();
public System.Drawing.Imaging.MetaHeader get_WmfHeader();
```

Public Methods

```
public boolean IsDisplay();
public boolean IsEmf();
public boolean IsEmfOrEmfPlus();
public boolean IsEmfPlus();
public boolean IsEmfPlusDual();
public boolean IsEmfPlusOnly();
public boolean IsWmf();
public boolean IsWmfPlaceable();
```

MetafileType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MetafileType
```

Members

Emf

EmfPlusDual

EmfPlusOnly

Invalid

Wmf

WmfPlaceable

MetaHeader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MetaHeader
   extends System.Object
```

Public Constructors

```
public MetaHeader();
```

Public Properties

```
public short get_HeaderSize();
public void set_HeaderSize(
   short value
);
public int get_MaxRecord();
public void set_MaxRecord(
   int value
);
public short get_NoObjects();
public void set_NoObjects(
   short value
);
public short get_NoParameters();
public void set_NoParameters(
   short value
);
public int get_Size();
public void set_Size(
   int value
);
public short get_Type();
public void set_Type(
   short value
);
public short get_Version();
public void set_Version(
   short value
);
```

PaletteFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PaletteFlags
```

Members

GrayScale

Halftone

HasAlpha

PixelFormat Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PixelFormat
```

Members

Alpha

Canonical

DontCare

Extended

Format16bppArgb1555

Format16bppGrayScale

Format16bppRgb555

Format16bppRgb565

Format1bppIndexed

Format24bppRgb

Format32bppArgb

Format32bppPArgb

Format32bppRgb

Format48bppRgb

Format4bppIndexed

Format64bppArgb

Format64bppPArgb

Format8bppIndexed

Gdi

Indexed

Max

PAlpha

Undefined

PlayRecordCallback Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void PlayRecordCallback(
   System.Drawing.Imaging.EmfPlusRecordType recordType,
   int flags,
   int dataSize,
   System.IntPtr recordData
);
```

PropertyItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PropertyItem
   extends System.Object
```

Public Properties

```
public int get_Id();
public void set_Id(
   int value
);
public int get_Len();
public void set_Len(
   int value
);
public short get_Type();
public void set_Type(
   short value
);
public ubyte[] get_Value();
public void set_Value(
   ubyte[] value
);
```

WmfPlaceableFileHeader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WmfPlaceableFileHeader
   extends System.Object
```

Public Constructors

```
public WmfPlaceableFileHeader();
```

Public Properties

```
public short get_BboxBottom();
public void set_BboxBottom(
   short value
);
public short get_BboxLeft();
public void set_BboxLeft(
   short value
);
public short get_BboxRight();
public void set_BboxRight(
   short value
);
public short get_BboxTop();
public void set_BboxTop(
   short value
);
public short get_Checksum();
public void set_Checksum(
   short value
);
public short get_Hmf();
public void set_Hmf(
   short value
);
public short get_Inch();
public void set_Inch(
   short value
);
public int get_Key();
public void set_Key(
   int value
);
public int get_Reserved();
public void set_Reserved(
   int value
);
```

System.Drawing.Printing Namespace (Visual J# Syntax)

Classes

InvalidPrinterException

Margins

MarginsConverter

PageSettings

PaperSize

PaperSource

PreviewPageInfo

PreviewPrintController

PrintController

PrintDocument

PrinterResolution

PrinterSettings

PrinterSettings.PaperSizeCollection

PrinterSettings.PaperSourceCollection

PrinterSettings.PrinterResolutionCollection

PrinterSettings.StringCollection

PrinterUnitConvert

PrintEventArgs

PrintingPermission

PrintingPermissionAttribute

PrintPageEventArgs

QueryPageSettingsEventArgs

StandardPrintController

Delegates

PrintEventHandler

PrintPageEventHandler

QueryPageSettingsEventHandler

Enumerations

Duplex

PaperKind

PaperSourceKind

PrinterResolutionKind

PrinterUnit

PrintingPermissionLevel

PrintRange

Duplex Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum Duplex
```

Members

Default

Horizontal

Simplex

Vertical

InvalidPrinterException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InvalidPrinterException
   extends System.SystemException
```

Public Constructors

```
public InvalidPrinterException(
   System.Drawing.Printing.PrinterSettings settings
);
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected InvalidPrinterException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Margins Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Drawi
ng.Printing.MarginsConverter, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyT
oken=b03f5f7f11d50a3a") */
public class Margins
   extends System.Object
   implements System.ICloneable
```

Public Constructors

```
public Margins();
public Margins(
   int left,
   int right,
   int top,
   int bottom
);
```

Public Properties

```
public int get_Bottom();
public void set_Bottom(
   int value
);
public int get_Left();
public void set_Left(
   int value
);
public int get_Right();
public void set_Right(
   int value
);
public int get_Top();
public void set_Top(
   int value
);
```

Public Methods

```
public final System.Object Clone();
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public System.String ToString();
```

MarginsConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MarginsConverter
   extends System.ComponentModel.ExpandableObjectConverter
```

Public Constructors

```
public MarginsConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.Object CreateInstance(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Collections.IDictionary propertyValues
);
public boolean GetCreateInstanceSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

PageSettings Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class PageSettings
   extends System.Object
   implements System.ICloneable
```

Public Constructors

```
public PageSettings();
public PageSettings(
   System.Drawing.Printing.PrinterSettings printerSettings
);
```

Public Properties

```
public System.Drawing.Rectangle get_Bounds();
public boolean get_Color();
public void set_Color(
   boolean value
);
public boolean get_Landscape();
public void set_Landscape(
   boolean value
);
public System.Drawing.Printing.Margins get_Margins();
public void set_Margins(
   System.Drawing.Printing.Margins value
);
public System.Drawing.Printing.PaperSize get_PaperSize();
public void set_PaperSize(
   System.Drawing.Printing.PaperSize value
);
public System.Drawing.Printing.PaperSource get_PaperSource();
public void set_PaperSource(
   System.Drawing.Printing.PaperSource value
);
public System.Drawing.Printing.PrinterResolution get_PrinterResolution();
public void set_PrinterResolution(
   System.Drawing.Printing.PrinterResolution value
);
public System.Drawing.Printing.PrinterSettings get_PrinterSettings();
public void set_PrinterSettings(
   System.Drawing.Printing.PrinterSettings value
);
```

Public Methods

```
public final System.Object Clone();
public void CopyToHdevmode(
   System.IntPtr hdevmode
);
public void SetHdevmode(
   System.IntPtr hdevmode
);
```

```
public System.String ToString();
```

PaperKind Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PaperKind
```

Members

A2

A3

A3Extra

A3ExtraTransverse

A3Rotated

A3Transverse

A4

A4Extra

A4Plus

A4Rotated

A4Small

A4Transverse

A5

A5Extra

A5Rotated

A5Transverse

A6

A6Rotated

APlus

B4

B4Envelope

B4JisRotated

B5

B5Envelope

B5Extra

B5JisRotated

B5Transverse

B6Envelope

B6Jis

B6JisRotated

BPlus

C3Envelope

C4Envelope

C5Envelope

C65Envelope

C6Envelope

CSheet

Custom

DLEnvelope

DSheet

ESheet

Executive

Folio

GermanLegalFanfold

GermanStandardFanfold

InviteEnvelope

IsoB4

ItalyEnvelope

JapaneseDoublePostcard

JapaneseDoublePostcardRotated

JapaneseEnvelopeChouNumber3

JapaneseEnvelopeChouNumber3Rotated

JapaneseEnvelopeChouNumber4

JapaneseEnvelopeChouNumber4Rotated

JapaneseEnvelopeKakuNumber2

JapaneseEnvelopeKakuNumber2Rotated

JapaneseEnvelopeKakuNumber3

JapaneseEnvelopeKakuNumber3Rotated

JapaneseEnvelopeYouNumber4

JapaneseEnvelopeYouNumber4Rotated

JapanesePostcard

JapanesePostcardRotated

Ledger

Legal

LegalExtra

Letter

LetterExtra

LetterExtraTransverse

LetterPlus

LetterRotated

LetterSmall

LetterTransverse

MonarchEnvelope

Note

Number10Envelope

Number11Envelope

Number12Envelope

Number14Envelope

Number9Envelope

PersonalEnvelope

Prc16K

Prc16KRotated

Prc32K

Prc32KBig

Prc32KBigRotated

Prc32KRotated

PrcEnvelopeNumber1

PrcEnvelopeNumber10

PrcEnvelopeNumber10Rotated

PrcEnvelopeNumber1Rotated

PrcEnvelopeNumber2

PrcEnvelopeNumber2Rotated

PrcEnvelopeNumber3

PrcEnvelopeNumber3Rotated

PrcEnvelopeNumber4

PrcEnvelopeNumber4Rotated

PrcEnvelopeNumber5

PrcEnvelopeNumber5Rotated

PrcEnvelopeNumber6

PrcEnvelopeNumber6Rotated

PrcEnvelopeNumber7

PrcEnvelopeNumber7Rotated

PrcEnvelopeNumber8

PrcEnvelopeNumber8Rotated

PrcEnvelopeNumber9

PrcEnvelopeNumber9Rotated

Quarto

Standard10x11

Standard10x14

Standard11x17

Standard12x11

Standard15x11

Standard9x11

Statement

Tabloid

TabloidExtra

USStandardFanfold

PaperSize Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PaperSize
   extends System.Object
```

Public Constructors

```
public PaperSize(
   System.String name,
   int width,
   int height
);
```

Public Properties

```
public int get_Height();
public void set_Height(
   int value
);
public System.Drawing.Printing.PaperKind get_Kind();
public System.String get_PaperName();
public void set_PaperName(
   System.String value
);
public int get_Width();
public void set_Width(
   int value
);
```

Public Methods

```
public System.String ToString();
```

PaperSource Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PaperSource
   extends System.Object
```

Public Properties

```
public System.Drawing.Printing.PaperSourceKind get_Kind();
public System.String get_SourceName();
```

Public Methods

```
public System.String ToString();
```

PaperSourceKind Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PaperSourceKind
```

Members

AutomaticFeed

Cassette

Custom

Envelope

FormSource

LargeCapacity

LargeFormat

Lower

Manual

ManualFeed

Middle

SmallFormat

TractorFeed

Upper

PreviewPageInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PreviewPageInfo
   extends System.Object
```

Public Constructors

```
public PreviewPageInfo(
   System.Drawing.Image image,
   System.Drawing.Size physicalSize
);
```

Public Properties

```
public System.Drawing.Image get_Image();
public System.Drawing.Size get_PhysicalSize();
```

PreviewPrintController Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PreviewPrintController
   extends System.Drawing.Printing.PrintController
```

Public Constructors

```
public PreviewPrintController();
```

Public Properties

```
public boolean get_UseAntiAlias();
public void set_UseAntiAlias(
   boolean value
);
```

Public Methods

```
public System.Drawing.Printing.PreviewPageInfo[] GetPreviewPageInfo();
public void OnEndPage(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintPageEventArgs e
);
public void OnEndPrint(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintEventArgs e
);
public System.Drawing.Graphics OnStartPage(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintPageEventArgs e
);
public void OnStartPrint(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintEventArgs e
);
```

PrintController Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class PrintController
   extends System.Object
```

Public Constructors

```
public PrintController();
```

Public Methods

```
public void OnEndPage(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintPageEventArgs e
);
public void OnEndPrint(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintEventArgs e
);
public System.Drawing.Graphics OnStartPage(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintPageEventArgs e
);
public void OnStartPrint(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintEventArgs e
);
```

PrintDocument Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemFilterAttribute(FilterString = "System.Drawin
g.Printing", FilterType = Allow) */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "DocumentName") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "PrintPage") */
public class PrintDocument
   extends System.ComponentModel.Component
```

Public Constructors

```
public PrintDocument();
```

Public Properties

```
public System.Drawing.Printing.PageSettings get_DefaultPageSettings();
public void set_DefaultPageSettings(
   System.Drawing.Printing.PageSettings value
);
public System.String get_DocumentName();
public void set_DocumentName(
   System.String value
);
public System.Drawing.Printing.PrintController get_PrintController();
public void set_PrintController(
   System.Drawing.Printing.PrintController value
);
public System.Drawing.Printing.PrinterSettings get_PrinterSettings();
public void set_PrinterSettings(
   System.Drawing.Printing.PrinterSettings value
);
```

Public Methods

```
public void Print();
public System.String ToString();
```

Public Events

```
public void add_BeginPrint(
   System.Drawing.Printing.PrintEventHandler value
);
public void remove_BeginPrint(
   System.Drawing.Printing.PrintEventHandler value
);
public void add_EndPrint(
   System.Drawing.Printing.PrintEventHandler value
);
public void remove_EndPrint(
   System.Drawing.Printing.PrintEventHandler value
);
public void add_PrintPage(
   System.Drawing.Printing.PrintPageEventHandler value
```

```
);
public void remove_PrintPage(
   System.Drawing.Printing.PrintPageEventHandler value
);
public void add_QueryPageSettings(
   System.Drawing.Printing.QueryPageSettingsEventHandler value
);
public void remove_QueryPageSettings(
   System.Drawing.Printing.QueryPageSettingsEventHandler value
);
```

Protected Methods

```
protected void OnBeginPrint(
   System.Drawing.Printing.PrintEventArgs e
);
protected void OnEndPrint(
   System.Drawing.Printing.PrintEventArgs e
);
protected void OnPrintPage(
   System.Drawing.Printing.PrintPageEventArgs e
);
protected void OnQueryPageSettings(
   System.Drawing.Printing.QueryPageSettingsEventArgs e
);
```

PrinterResolution Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PrinterResolution
   extends System.Object
```

Public Properties

```
public System.Drawing.Printing.PrinterResolutionKind get_Kind();
public int get_X();
public int get_Y();
```

Public Methods

```
public System.String ToString();
```

PrinterResolutionKind Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PrinterResolutionKind
```

Members

Custom

Draft

High

Low

Medium

PrinterSettings Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class PrinterSettings
   extends System.Object
   implements System.ICloneable
```

Public Constructors

```
public PrinterSettings();
```

Public Properties

```
public boolean get_CanDuplex();
public boolean get_Collate();
public void set_Collate(
   boolean value
);
public short get_Copies();
public void set_Copies(
   short value
);
public System.Drawing.Printing.PageSettings get_DefaultPageSettings();
public System.Drawing.Printing.Duplex get_Duplex();
public void set_Duplex(
   System.Drawing.Printing.Duplex value
);
public int get_FromPage();
public void set_FromPage(
   int value
);
public static System.Drawing.Printing.PrinterSettings+StringCollection get_InstalledPrinters(
);
public boolean get_IsDefaultPrinter();
public boolean get_IsPlotter();
public boolean get_IsValid();
public int get_LandscapeAngle();
public int get_MaximumCopies();
public int get_MaximumPage();
public void set_MaximumPage(
   int value
);
public int get_MinimumPage();
public void set_MinimumPage(
   int value
);
public System.Drawing.Printing.PrinterSettings+PaperSizeCollection get_PaperSizes();
public System.Drawing.Printing.PrinterSettings+PaperSourceCollection get_PaperSources();
public System.String get_PrinterName();
public void set_PrinterName(
   System.String value
);
public System.Drawing.Printing.PrinterSettings+PrinterResolutionCollection get_PrinterResolut
ions();
public System.Drawing.Printing.PrintRange get_PrintRange();
public void set_PrintRange(
```

```
   System.Drawing.Printing.PrintRange value
);
public boolean get_PrintToFile();
public void set_PrintToFile(
   boolean value
);
public boolean get_SupportsColor();
public int get_ToPage();
public void set_ToPage(
   int value
);
```

Public Methods

```
public final System.Object Clone();
public System.Drawing.Graphics CreateMeasurementGraphics();
public System.IntPtr GetHdevmode();
public System.IntPtr GetHdevmode(
   System.Drawing.Printing.PageSettings pageSettings
);
public System.IntPtr GetHdevnames();
public void SetHdevmode(
   System.IntPtr hdevmode
);
public void SetHdevnames(
   System.IntPtr hdevnames
);
public System.String ToString();
```

PrinterSettings.PaperSizeCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class PaperSizeCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public PaperSizeCollection(
   System.Drawing.Printing.PaperSize[] array
);
```

Public Properties

```
public int get_Count();
public System.Drawing.Printing.PaperSize get_Item(
   int index
);
```

Public Methods

```
public System.Collections.IEnumerator GetEnumerator();
```

PrinterSettings.PaperSourceCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class PaperSourceCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public PaperSourceCollection(
   System.Drawing.Printing.PaperSource[] array
);
```

Public Properties

```
public int get_Count();
public System.Drawing.Printing.PaperSource get_Item(
   int index
);
```

Public Methods

```
public System.Collections.IEnumerator GetEnumerator();
```

PrinterSettings.PrinterResolutionCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class PrinterResolutionCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public PrinterResolutionCollection(
   System.Drawing.Printing.PrinterResolution[] array
);
```

Public Properties

```
public int get_Count();
public System.Drawing.Printing.PrinterResolution get_Item(
   int index
);
```

Public Methods

```
public System.Collections.IEnumerator GetEnumerator();
```

PrinterSettings.StringCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class StringCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public StringCollection(
   System.String[] array
);
```

Public Properties

```
public int get_Count();
public System.String get_Item(
   int index
);
```

Public Methods

```
public System.Collections.IEnumerator GetEnumerator();
```

PrinterUnit Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PrinterUnit
```

Members

Display

HundredthsOfAMillimeter

TenthsOfAMillimeter

ThousandthsOfAnInch

PrinterUnitConvert Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PrinterUnitConvert
   extends System.Object
```

Public Methods

```
public static double Convert(
   double value,
   System.Drawing.Printing.PrinterUnit fromUnit,
   System.Drawing.Printing.PrinterUnit toUnit
);
public static int Convert(
   int value,
   System.Drawing.Printing.PrinterUnit fromUnit,
   System.Drawing.Printing.PrinterUnit toUnit
);
public static System.Drawing.Point Convert(
   System.Drawing.Point value,
   System.Drawing.Printing.PrinterUnit fromUnit,
   System.Drawing.Printing.PrinterUnit toUnit
);
public static System.Drawing.Printing.Margins Convert(
   System.Drawing.Printing.Margins value,
   System.Drawing.Printing.PrinterUnit fromUnit,
   System.Drawing.Printing.PrinterUnit toUnit
);
public static System.Drawing.Rectangle Convert(
   System.Drawing.Rectangle value,
   System.Drawing.Printing.PrinterUnit fromUnit,
   System.Drawing.Printing.PrinterUnit toUnit
);
public static System.Drawing.Size Convert(
   System.Drawing.Size value,
   System.Drawing.Printing.PrinterUnit fromUnit,
   System.Drawing.Printing.PrinterUnit toUnit
);
```

PrintEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PrintEventArgs
   extends System.ComponentModel.CancelEventArgs
```

Public Constructors

```
public PrintEventArgs();
```

PrintEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void PrintEventHandler(
   System.Object sender,
   System.Drawing.Printing.PrintEventArgs e
);
```

PrintingPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class PrintingPermission
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Constructors

```
public PrintingPermission(
   System.Drawing.Printing.PrintingPermissionLevel printingLevel
);
public PrintingPermission(
   System.Security.Permissions.PermissionState state
);
```

Public Properties

```
public System.Drawing.Printing.PrintingPermissionLevel get_Level();
public void set_Level(
   System.Drawing.Printing.PrintingPermissionLevel value
);
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement esd
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public final boolean IsUnrestricted();
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

PrintingPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = true, Inherited
= true) */
public class PrintingPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public PrintingPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.Drawing.Printing.PrintingPermissionLevel get_Level();
public void set_Level(
   System.Drawing.Printing.PrintingPermissionLevel value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

PrintingPermissionLevel Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PrintingPermissionLevel
```

Members

AllPrinting

DefaultPrinting

NoPrinting

SafePrinting

PrintPageEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PrintPageEventArgs
   extends System.EventArgs
```

Public Constructors

```
public PrintPageEventArgs(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle marginBounds,
   System.Drawing.Rectangle pageBounds,
   System.Drawing.Printing.PageSettings pageSettings
);
```

Public Properties

```
public boolean get_Cancel();
public void set_Cancel(
   boolean value
);
public System.Drawing.Graphics get_Graphics();
public boolean get_HasMorePages();
public void set_HasMorePages(
   boolean value
);
public System.Drawing.Rectangle get_MarginBounds();
public System.Drawing.Rectangle get_PageBounds();
public System.Drawing.Printing.PageSettings get_PageSettings();
```

PrintPageEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void PrintPageEventHandler(
   System.Object sender,
   System.Drawing.Printing.PrintPageEventArgs e
);
```

PrintRange Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PrintRange
```

Members

AllPages

Selection

SomePages

QueryPageSettingsEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class QueryPageSettingsEventArgs
   extends System.Drawing.Printing.PrintEventArgs
```

Public Constructors

```
public QueryPageSettingsEventArgs(
   System.Drawing.Printing.PageSettings pageSettings
);
```

Public Properties

```
public System.Drawing.Printing.PageSettings get_PageSettings();
public void set_PageSettings(
   System.Drawing.Printing.PageSettings value
);
```

QueryPageSettingsEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void QueryPageSettingsEventHandler(
   System.Object sender,
   System.Drawing.Printing.QueryPageSettingsEventArgs e
);
```

StandardPrintController Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class StandardPrintController
   extends System.Drawing.Printing.PrintController
```

Public Constructors

```
public StandardPrintController();
```

Public Methods

```
public void OnEndPage(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintPageEventArgs e
);
public void OnEndPrint(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintEventArgs e
);
public System.Drawing.Graphics OnStartPage(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintPageEventArgs e
);
public void OnStartPrint(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintEventArgs e
);
```

System.Drawing.Text Namespace (Visual J# Syntax)

Classes

FontCollection

InstalledFontCollection

PrivateFontCollection

Enumerations

GenericFontFamilies

HotkeyPrefix

TextRenderingHint

FontCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class FontCollection
   extends System.Object
   implements System.IDisposable
```

Public Properties

```
public System.Drawing.FontFamily[] get_Families();
```

Public Methods

```
public final void Dispose();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

GenericFontFamilies Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum GenericFontFamilies
```

Members

Monospace

SansSerif

Serif

HotkeyPrefix Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum HotkeyPrefix
```

Members

Hide

None

Show

InstalledFontCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InstalledFontCollection
   extends System.Drawing.Text.FontCollection
```

Public Constructors

```
public InstalledFontCollection();
```

PrivateFontCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class PrivateFontCollection
   extends System.Drawing.Text.FontCollection
```

Public Constructors

```
public PrivateFontCollection();
```

Public Methods

```
public void AddFontFile(
   System.String filename
);
public void AddMemoryFont(
   System.IntPtr memory,
   int length
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

TextRenderingHint Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TextRenderingHint
```

Members

AntiAlias

AntiAliasGridFit

ClearTypeGridFit

SingleBitPerPixel

SingleBitPerPixelGridFit

SystemDefault

System.EnterpriseServices Namespace (Visual J# Syntax)

Classes

ApplicationAccessControlAttribute

ApplicationActivationAttribute

ApplicationIDAttribute

ApplicationNameAttribute

ApplicationQueuingAttribute

AutoCompleteAttribute

BYOT

ComponentAccessControlAttribute

COMTIIntrinsicsAttribute

ConstructionEnabledAttribute

ContextUtil

DescriptionAttribute

EventClassAttribute

EventTrackingEnabledAttribute

ExceptionClassAttribute

IISIntrinsicsAttribute

InterfaceQueuingAttribute

JustInTimeActivationAttribute

LoadBalancingSupportedAttribute

MustRunInClientContextAttribute

ObjectPoolingAttribute

PrivateComponentAttribute

RegistrationErrorInfo

RegistrationException

RegistrationHelper

RegistrationHelperTx

ResourcePool

SecureMethodAttribute

SecurityCallContext

SecurityCallers

SecurityIdentity

SecurityRoleAttribute

ServicedComponent

ServicedComponentException

SharedProperty

SharedPropertyGroup

SharedPropertyGroupManager

SynchronizationAttribute

TransactionAttribute

Interfaces

IRegistrationHelper

IRemoteDispatch

IServicedComponentInfo

ITransaction

Structures

BOID

XACTTRANSINFO

Delegates

ResourcePool.TransactionEndDelegate

Enumerations

AccessChecksLevelOption

ActivationOption

AuthenticationOption

ImpersonationLevelOption

InstallationFlags

PropertyLockMode

PropertyReleaseMode

SynchronizationOption

TransactionIsolationLevel

TransactionOption

TransactionVote

AccessChecksLevelOption Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum AccessChecksLevelOption
```

Members

Application

ApplicationComponent

ActivationOption Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ActivationOption
```

Members

Library

Server

ApplicationAccessControlAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class ApplicationAccessControlAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public ApplicationAccessControlAttribute();
public ApplicationAccessControlAttribute(
   boolean val
);
```

Public Properties

```
public System.EnterpriseServices.AccessChecksLevelOption get_AccessChecksLevel();
public void set_AccessChecksLevel(
   System.EnterpriseServices.AccessChecksLevelOption value
);
public System.EnterpriseServices.AuthenticationOption get_Authentication();
public void set_Authentication(
   System.EnterpriseServices.AuthenticationOption value
);
public System.EnterpriseServices.ImpersonationLevelOption get_ImpersonationLevel();
public void set_ImpersonationLevel(
   System.EnterpriseServices.ImpersonationLevelOption value
);
public boolean get_Value();
public void set_Value(
   boolean value
);
```

ApplicationActivationAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class ApplicationActivationAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public ApplicationActivationAttribute(
   System.EnterpriseServices.ActivationOption opt
);
```

Public Properties

```
public System.String get_SoapMailbox();
public void set_SoapMailbox(
   System.String value
);
public System.String get_SoapVRoot();
public void set_SoapVRoot(
   System.String value
);
public System.EnterpriseServices.ActivationOption get_Value();
```

ApplicationIDAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class ApplicationIDAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public ApplicationIDAttribute(
   System.String guid
);
```

Public Properties

```
public System.Guid get_Value();
```

ApplicationNameAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class ApplicationNameAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public ApplicationNameAttribute(
   System.String name
);
```

Public Properties

```
public System.String get_Value();
```

ApplicationQueuingAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class ApplicationQueuingAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public ApplicationQueuingAttribute();
```

Public Properties

```
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public int get_MaxListenerThreads();
public void set_MaxListenerThreads(
   int value
);
public boolean get_QueueListenerEnabled();
public void set_QueueListenerEnabled(
   boolean value
);
```

AuthenticationOption Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum AuthenticationOption
```

Members

Call

Connect

Default

Integrity

None

Packet

Privacy

AutoCompleteAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = true) */
public class AutoCompleteAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public AutoCompleteAttribute();
public AutoCompleteAttribute(
   boolean val
);
```

Public Properties

```
public boolean get_Value();
```

BOID Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class BOID
   extends System.ValueType
```

Public Fields

```
public ubyte[] rgb;
```

BYOT Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BYOT
   extends System.Object
```

Public Methods

```
public static System.Object CreateWithTipTransaction(
   System.String url,
   System.Type t
);
public static System.Object CreateWithTransaction(
   System.Object transaction,
   System.Type t
);
```

ComponentAccessControlAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class ComponentAccessControlAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public ComponentAccessControlAttribute();
public ComponentAccessControlAttribute(
   boolean val
);
```

Public Properties

```
public boolean get_Value();
```

COMTIIntrinsicsAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class COMTIIntrinsicsAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public COMTIIntrinsicsAttribute();
public COMTIIntrinsicsAttribute(
   boolean val
);
```

Public Properties

```
public boolean get_Value();
```

ConstructionEnabledAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class ConstructionEnabledAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public ConstructionEnabledAttribute();
public ConstructionEnabledAttribute(
   boolean val
);
```

Public Properties

```
public System.String get_Default();
public void set_Default(
   System.String value
);
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
```

ContextUtil Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ContextUtil
   extends System.Object
```

Public Properties

```
public static System.Guid get_ActivityId();
public static System.Guid get_ApplicationId();
public static System.Guid get_ApplicationInstanceId();
public static System.Guid get_ContextId();
public static boolean get_DeactivateOnReturn();
public static void set_DeactivateOnReturn(
   boolean value
);
public static boolean get_IsInTransaction();
public static boolean get_IsSecurityEnabled();
public static System.EnterpriseServices.TransactionVote get_MyTransactionVote();
public static void set_MyTransactionVote(
   System.EnterpriseServices.TransactionVote value
);
public static System.Guid get_PartitionId();
public static System.Object get_Transaction();
public static System.Guid get_TransactionId();
```

Public Methods

```
public static void DisableCommit();
public static void EnableCommit();
public static System.Object GetNamedProperty(
   System.String name
);
public static boolean IsCallerInRole(
   System.String role
);
public static void SetAbort();
public static void SetComplete();
```

DescriptionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Method, Interface, A
llowMultiple = false, Inherited = true) */
public class DescriptionAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public DescriptionAttribute(
   System.String desc
);
```

EventClassAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class EventClassAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public EventClassAttribute();
```

Public Properties

```
public boolean get_AllowInprocSubscribers();
public void set_AllowInprocSubscribers(
   boolean value
);
public boolean get_FireInParallel();
public void set_FireInParallel(
   boolean value
);
public System.String get_PublisherFilter();
public void set_PublisherFilter(
   System.String value
);
```

EventTrackingEnabledAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class EventTrackingEnabledAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public EventTrackingEnabledAttribute();
public EventTrackingEnabledAttribute(
   boolean val
);
```

Public Properties

```
public boolean get_Value();
```

ExceptionClassAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class ExceptionClassAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public ExceptionClassAttribute(
   System.String name
);
```

Public Properties

```
public System.String get_Value();
```

IISIntrinsicsAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class IISIntrinsicsAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public IISIntrinsicsAttribute();
public IISIntrinsicsAttribute(
   boolean val
);
```

Public Properties

```
public boolean get_Value();
```

ImpersonationLevelOption Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ImpersonationLevelOption
```

Members

Anonymous

Default

Delegate

Identify

Impersonate

InstallationFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum InstallationFlags
```

Members

Configure

ConfigureComponentsOnly

CreateTargetApplication

Default

ExpectExistingTypeLib

FindOrCreateTargetApplication

Install

ReconfigureExistingApplication

Register

ReportWarningsToConsole

InterfaceQueuingAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Interface, AllowMultiple = tru
e, Inherited = true) */
public class InterfaceQueuingAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public InterfaceQueuingAttribute();
public InterfaceQueuingAttribute(
   boolean enabled
);
```

Public Properties

```
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public System.String get_Interface();
public void set_Interface(
   System.String value
);
```

IRegistrationHelper Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "55e3ea25-55cb-4650-8887-
18e8d30bb4bc") */
public interface IRegistrationHelper
```

Public Methods

```
void InstallAssembly(
   System.String assembly,
   System.String& application,
   System.String& tlb,
   System.EnterpriseServices.InstallationFlags installFlags
);
void UninstallAssembly(
   System.String assembly,
   System.String application
);
```

IRemoteDispatch Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "6619a740-8154-43be-a186-
0319578e02db") */
public interface IRemoteDispatch
```

Public Methods

```
/** @attribute System.EnterpriseServices.AutoCompleteAttribute(Value = true) */
System.String RemoteDispatchAutoDone(
   System.String s
);
/** @attribute System.EnterpriseServices.AutoCompleteAttribute(Value = false) */
System.String RemoteDispatchNotAutoDone(
   System.String s
);
```

IServicedComponentInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "8165B19E-8D3A-4d0b-80C8-
97DE310DB583") */
public interface IServicedComponentInfo
```

Public Methods

```
void GetComponentInfo(
   int infoMask,
   System.String[]& infoArray
);
```

ITransaction Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "0FB15084-AF41-11CE-BD2B-
204C4F4F5020") */
public interface ITransaction
```

Public Methods

```
void Abort(
   System.EnterpriseServices.BOID& pboidReason,
   int fRetaining,
   int fAsync
);
void Commit(
   int fRetaining,
   int grfTC,
   int grfRM
);
void GetTransactionInfo(
   System.EnterpriseServices.XACTTRANSINFO& pinfo
);
```

JustInTimeActivationAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class JustInTimeActivationAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public JustInTimeActivationAttribute();
public JustInTimeActivationAttribute(
   boolean val
);
```

Public Properties

```
public boolean get_Value();
```

LoadBalancingSupportedAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class LoadBalancingSupportedAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public LoadBalancingSupportedAttribute();
public LoadBalancingSupportedAttribute(
   boolean val
);
```

Public Properties

```
public boolean get_Value();
```

MustRunInClientContextAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class MustRunInClientContextAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public MustRunInClientContextAttribute();
public MustRunInClientContextAttribute(
   boolean val
);
```

Public Properties

```
public boolean get_Value();
```

ObjectPoolingAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class ObjectPoolingAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public ObjectPoolingAttribute();
public ObjectPoolingAttribute(
   boolean enable
);
public ObjectPoolingAttribute(
   int minPoolSize,
   int maxPoolSize
);
public ObjectPoolingAttribute(
   boolean enable,
   int minPoolSize,
   int maxPoolSize
);
```

Public Properties

```
public int get_CreationTimeout();
public void set_CreationTimeout(
   int value
);
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public int get_MaxPoolSize();
public void set_MaxPoolSize(
   int value
);
public int get_MinPoolSize();
public void set_MinPoolSize(
   int value
);
```

Public Methods

```
public final boolean AfterSaveChanges(
   System.Collections.Hashtable info
);
public final boolean Apply(
   System.Collections.Hashtable info
);
public final boolean IsValidTarget(
   System.String s
);
```

PrivateComponentAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class PrivateComponentAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public PrivateComponentAttribute();
```

PropertyLockMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public enum PropertyLockMode
```

Members

Method

SetGet

PropertyReleaseMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public enum PropertyReleaseMode
```

Members

Process

Standard

RegistrationErrorInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RegistrationErrorInfo
   extends System.Object
```

Public Properties

```
public int get_ErrorCode();
public System.String get_ErrorString();
public System.String get_MajorRef();
public System.String get_MinorRef();
public System.String get_Name();
```

RegistrationException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RegistrationException
   extends System.SystemException
```

Public Constructors

```
public RegistrationException(
   System.String msg
);
```

Public Properties

```
public System.EnterpriseServices.RegistrationErrorInfo[] get_ErrorInfo();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext ctx
);
```

RegistrationHelper Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "89a86e7b-c229-4008-9baa-
2f5c8411d7e0") */
public class RegistrationHelper
   extends System.MarshalByRefObject
   implements System.EnterpriseServices.IRegistrationHelper, System.EnterpriseServices.Thunk.
IThunkInstallation
```

Public Constructors

```
public RegistrationHelper();
```

Public Methods

```
public final void InstallAssembly(
   System.String assembly,
   System.String& application,
   System.String& tlb,
   System.EnterpriseServices.InstallationFlags installFlags
);
public void InstallAssembly(
   System.String assembly,
   System.String& application,
   System.String partition,
   System.String& tlb,
   System.EnterpriseServices.InstallationFlags installFlags
);
public final void UninstallAssembly(
   System.String assembly,
   System.String application
);
public void UninstallAssembly(
   System.String assembly,
   System.String application,
   System.String partition
);
```

RegistrationHelperTx Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.EnterpriseServices.TransactionAttribute(Value = RequiresNew, Isolation
= Serializable, Timeout = -1) */
public class RegistrationHelperTx
   extends System.EnterpriseServices.ServicedComponent
```

Public Constructors

```
public RegistrationHelperTx();
```

Public Methods

```
public void InstallAssembly(
   System.String assembly,
   System.String& application,
   System.String& tlb,
   System.EnterpriseServices.InstallationFlags installFlags,
   System.Object sync
);
public void InstallAssembly(
   System.String assembly,
   System.String& application,
   System.String partition,
   System.String& tlb,
   System.EnterpriseServices.InstallationFlags installFlags,
   System.Object sync
);
public boolean IsInTransaction();
public void UninstallAssembly(
   System.String assembly,
   System.String application,
   System.Object sync
);
public void UninstallAssembly(
   System.String assembly,
   System.String application,
   System.String partition,
   System.Object sync
);
```

Protected Methods

```
protected void Activate();
protected void Deactivate();
```

ResourcePool Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ResourcePool
   extends System.Object
   implements System.EnterpriseServices.IObjPool
```

Public Constructors

```
public ResourcePool(
   System.EnterpriseServices.ResourcePool+TransactionEndDelegate cb
);
```

Public Methods

```
public System.Object GetResource();
public boolean PutResource(
   System.Object resource
);
```

ResourcePool.TransactionEndDelegate Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void TransactionEndDelegate(
   System.Object resource
);
```

SecureMethodAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Method, AllowMultipl
e = false, Inherited = true) */
public class SecureMethodAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public SecureMethodAttribute();
```

SecurityCallContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SecurityCallContext
   extends System.Object
```

Public Properties

```
public System.EnterpriseServices.SecurityCallers get_Callers();
public static System.EnterpriseServices.SecurityCallContext get_CurrentCall();
public System.EnterpriseServices.SecurityIdentity get_DirectCaller();
public boolean get_IsSecurityEnabled();
public int get_MinAuthenticationLevel();
public int get_NumCallers();
public System.EnterpriseServices.SecurityIdentity get_OriginalCaller();
```

Public Methods

```
public boolean IsCallerInRole(
   System.String role
);
public boolean IsUserInRole(
   System.String user,
   System.String role
);
```

SecurityCallers Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SecurityCallers
   extends System.Object
   implements System.Collections.IEnumerable
```

Public Properties

```
public int get_Count();
public System.EnterpriseServices.SecurityIdentity get_Item(
   int idx
);
```

Public Methods

```
public final System.Collections.IEnumerator GetEnumerator();
```

SecurityIdentity Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SecurityIdentity
   extends System.Object
```

Public Properties

```
public System.String get_AccountName();
public System.EnterpriseServices.AuthenticationOption get_AuthenticationLevel();
public int get_AuthenticationService();
public System.EnterpriseServices.ImpersonationLevelOption get_ImpersonationLevel();
```

SecurityRoleAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Method, Interface, A
llowMultiple = true, Inherited = true) */
public class SecurityRoleAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public SecurityRoleAttribute(
   System.String role
);
public SecurityRoleAttribute(
   System.String role,
   boolean everyone
);
```

Public Properties

```
public System.String get_Description();
public void set_Description(
   System.String value
);
public System.String get_Role();
public void set_Role(
   System.String value
);
public boolean get_SetEveryoneAccess();
public void set_SetEveryoneAccess(
   boolean value
);
```

ServicedComponent Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.EnterpriseServices.ServicedComponentProxyAttribute() */
public abstract class ServicedComponent
   extends System.ContextBoundObject
   implements System.EnterpriseServices.IRemoteDispatch, System.IDisposable, System.Enterpris
eServices.IManagedObject, System.EnterpriseServices.IServicedComponentInfo
```

Public Constructors

```
public ServicedComponent();
```

Public Methods

```
public final void Dispose();
public static void DisposeObject(
   System.EnterpriseServices.ServicedComponent sc
);
```

Protected Methods

```
protected void Activate();
protected boolean CanBePooled();
protected void Construct(
   System.String s
);
protected void Deactivate();
protected void Dispose(
   boolean disposing
);
```

ServicedComponentException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ServicedComponentException
   extends System.SystemException
```

Public Constructors

```
public ServicedComponentException();
public ServicedComponentException(
   System.String message
);
public ServicedComponentException(
   System.String message,
   System.Exception innerException
);
```

SharedProperty Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class SharedProperty
   extends System.Object
```

Public Properties

```
public System.Object get_Value();
public void set_Value(
   System.Object value
);
```

SharedPropertyGroup Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class SharedPropertyGroup
   extends System.Object
```

Public Methods

```
public System.EnterpriseServices.SharedProperty CreateProperty(
   System.String name,
   boolean fExists
);
public System.EnterpriseServices.SharedProperty CreatePropertyByPosition(
   int position,
   boolean fExists
);
public System.EnterpriseServices.SharedProperty Property(
   System.String name
);
public System.EnterpriseServices.SharedProperty PropertyByPosition(
   int position
);
```

SharedPropertyGroupManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class SharedPropertyGroupManager
   extends System.Object
   implements System.Collections.IEnumerable
```

Public Constructors

```
public SharedPropertyGroupManager();
```

Public Methods

```
public System.EnterpriseServices.SharedPropertyGroup CreatePropertyGroup(
   System.String name,
   System.EnterpriseServices.PropertyLockMode& dwIsoMode,
   System.EnterpriseServices.PropertyReleaseMode& dwRelMode,
   boolean fExist
);
public final System.Collections.IEnumerator GetEnumerator();
public System.EnterpriseServices.SharedPropertyGroup Group(
   System.String name
);
```

SynchronizationAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class SynchronizationAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public SynchronizationAttribute();
public SynchronizationAttribute(
   System.EnterpriseServices.SynchronizationOption val
);
```

Public Properties

```
public System.EnterpriseServices.SynchronizationOption get_Value();
```

SynchronizationOption Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SynchronizationOption
```

Members

Disabled

NotSupported

Required

RequiresNew

Supported

TransactionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class TransactionAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public TransactionAttribute();
public TransactionAttribute(
   System.EnterpriseServices.TransactionOption val
);
```

Public Properties

```
public System.EnterpriseServices.TransactionIsolationLevel get_Isolation();
public void set_Isolation(
   System.EnterpriseServices.TransactionIsolationLevel value
);
public int get_Timeout();
public void set_Timeout(
   int value
);
public System.EnterpriseServices.TransactionOption get_Value();
```

TransactionIsolationLevel Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TransactionIsolationLevel
```

Members

Any

ReadCommitted

ReadUncommitted

RepeatableRead

Serializable

TransactionOption Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TransactionOption
```

Members

Disabled

NotSupported

Required

RequiresNew

Supported

TransactionVote Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public enum TransactionVote
```

Members

Abort

Commit

XACTTRANSINFO Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class XACTTRANSINFO
   extends System.ValueType
```

Public Fields

```
public int grfRMSupported;
public int grfRMSupportedRetaining;
public int grfTCSupported;
public int grfTCSupportedRetaining;
public int isoFlags;
public int isoLevel;
public System.EnterpriseServices.BOID uow;
```

System.EnterpriseServices.CompensatingResourceManager Namespace (Visual J# Syntax)

Classes

ApplicationCrmEnabledAttribute

Clerk

ClerkInfo

ClerkMonitor

Compensator

LogRecord

Enumerations

CompensatorOptions

LogRecordFlags

TransactionState

ApplicationCrmEnabledAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ProgIdAttribute(Value = "System.EnterpriseServi
ces.Crm.ApplicationCrmEnabledAttribute") */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class ApplicationCrmEnabledAttribute
   extends System.Attribute
   implements System.EnterpriseServices.IConfigurationAttribute
```

Public Constructors

```
public ApplicationCrmEnabledAttribute();
public ApplicationCrmEnabledAttribute(
   boolean val
);
```

Public Properties

```
public boolean get_Value();
```

Clerk Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Clerk
   extends System.Object
```

Public Constructors

```
public Clerk(
   System.String compensator,
   System.String description,
   System.EnterpriseServices.CompensatingResourceManager.CompensatorOptions flags
);
public Clerk(
   System.Type compensator,
   System.String description,
   System.EnterpriseServices.CompensatingResourceManager.CompensatorOptions flags
);
```

Public Properties

```
public int get_LogRecordCount();
public System.String get_TransactionUOW();
```

Public Methods

```
public void ForceLog();
public void ForceTransactionToAbort();
public void ForgetLogRecord();
public void WriteLogRecord(
   System.Object record
);
```

Protected Methods

```
protected void Finalize();
```

ClerkInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ClerkInfo
   extends System.Object
```

Public Properties

```
public System.String get_ActivityId();
public System.EnterpriseServices.CompensatingResourceManager.Clerk get_Clerk();
public System.String get_Compensator();
public System.String get_Description();
public System.String get_InstanceId();
public System.String get_TransactionUOW();
```

Protected Methods

```
protected void Finalize();
```

ClerkMonitor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ClerkMonitor
   extends System.Object
   implements System.Collections.IEnumerable
```

Public Constructors

```
public ClerkMonitor();
```

Public Properties

```
public int get_Count();
public System.EnterpriseServices.CompensatingResourceManager.ClerkInfo get_Item(
   int index
);
public System.EnterpriseServices.CompensatingResourceManager.ClerkInfo get_Item(
   System.String index
);
```

Public Methods

```
public final System.Collections.IEnumerator GetEnumerator();
public void Populate();
```

Protected Methods

```
protected void Finalize();
```

Compensator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Compensator
   extends System.EnterpriseServices.ServicedComponent
```

Public Constructors

```
public Compensator();
```

Public Properties

```
public System.EnterpriseServices.CompensatingResourceManager.Clerk get_Clerk();
```

Public Methods

```
public boolean AbortRecord(
   System.EnterpriseServices.CompensatingResourceManager.LogRecord rec
);
public void BeginAbort(
   boolean fRecovery
);
public void BeginCommit(
   boolean fRecovery
);
public void BeginPrepare();
public boolean CommitRecord(
   System.EnterpriseServices.CompensatingResourceManager.LogRecord rec
);
public void EndAbort();
public void EndCommit();
public boolean EndPrepare();
public boolean PrepareRecord(
   System.EnterpriseServices.CompensatingResourceManager.LogRecord rec
);
```

CompensatorOptions Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum CompensatorOptions
```

Members

AbortPhase

AllPhases

CommitPhase

FailIfInDoubtsRemain

PreparePhase

LogRecord Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LogRecord
   extends System.Object
```

Public Properties

```
public System.EnterpriseServices.CompensatingResourceManager.LogRecordFlags get_Flags();
public System.Object get_Record();
public int get_Sequence();
```

LogRecordFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum LogRecordFlags
```

Members

ForgetTarget

ReplayInProgress

WrittenDuringAbort

WrittenDuringCommit

WrittenDuringPrepare

WrittenDuringReplay

WrittenDurringRecovery

TransactionState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TransactionState
```

Members

Aborted

Active

Committed

Indoubt

System.Globalization Namespace (Visual J# Syntax)

Classes

Calendar

CompareInfo

CultureInfo

DateTimeFormatInfo

DaylightTime

GregorianCalendar

HebrewCalendar

HijriCalendar

JapaneseCalendar

JulianCalendar

KoreanCalendar

NumberFormatInfo

RegionInfo

SortKey

StringInfo

TaiwanCalendar

TextElementEnumerator

TextInfo

ThaiBuddhistCalendar

Enumerations

CalendarWeekRule

CompareOptions

CultureTypes

DateTimeStyles

GregorianCalendarTypes

NumberStyles

UnicodeCategory

Calendar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class Calendar
   extends System.Object
```

Public Properties

```
public abstract int[] get_Eras();
public int get_TwoDigitYearMax();
public void set_TwoDigitYearMax(
   int value
);
```

Public Methods

```
public System.DateTime AddDays(
   System.DateTime time,
   int days
);
public System.DateTime AddHours(
   System.DateTime time,
   int hours
);
public System.DateTime AddMilliseconds(
   System.DateTime time,
   double milliseconds
);
public System.DateTime AddMinutes(
   System.DateTime time,
   int minutes
);
public abstract System.DateTime AddMonths(
   System.DateTime time,
   int months
);
public System.DateTime AddSeconds(
   System.DateTime time,
   int seconds
);
public System.DateTime AddWeeks(
   System.DateTime time,
   int weeks
);
public abstract System.DateTime AddYears(
   System.DateTime time,
   int years
);
public abstract int GetDayOfMonth(
   System.DateTime time
);
public abstract System.DayOfWeek GetDayOfWeek(
   System.DateTime time
);
public abstract int GetDayOfYear(
   System.DateTime time
);
```

```
public int GetDaysInMonth(
   int year,
   int month
);
public abstract int GetDaysInMonth(
   int year,
   int month,
   int era
);
public int GetDaysInYear(
   int year
);
public abstract int GetDaysInYear(
   int year,
   int era
);
public abstract int GetEra(
   System.DateTime time
);
public int GetHour(
   System.DateTime time
);
public double GetMilliseconds(
   System.DateTime time
);
public int GetMinute(
   System.DateTime time
);
public abstract int GetMonth(
   System.DateTime time
);
public int GetMonthsInYear(
   int year
);
public abstract int GetMonthsInYear(
   int year,
   int era
);
public int GetSecond(
   System.DateTime time
);
public int GetWeekOfYear(
   System.DateTime time,
   System.Globalization.CalendarWeekRule rule,
   System.DayOfWeek firstDayOfWeek
);
public abstract int GetYear(
   System.DateTime time
);
public boolean IsLeapDay(
   int year,
   int month,
   int day
);
public abstract boolean IsLeapDay(
   int year,
   int month,
   int day,
   int era
);
public boolean IsLeapMonth(
   int year,
   int month
);
public abstract boolean IsLeapMonth(
   int year,
   int month,
   int era
```

```
);
public boolean IsLeapYear(
   int year
);
public abstract boolean IsLeapYear(
   int year,
   int era
);
public System.DateTime ToDateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second,
   int millisecond
);
public abstract System.DateTime ToDateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second,
   int millisecond,
   int era
);
public int ToFourDigitYear(
   int year
);
```

Public Fields

```
public static int CurrentEra;
```

Protected Constructors

```
protected Calendar();
```

CalendarWeekRule Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CalendarWeekRule
```

Members

FirstDay

FirstFourDayWeek

FirstFullWeek

CompareInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class CompareInfo
   extends System.Object
   implements System.Runtime.Serialization.IDeserializationCallback
```

Public Properties

```
public int get_LCID();
```

Public Methods

```
public int Compare(
   System.String string1,
   System.String string2
);
public int Compare(
   System.String string1,
   System.String string2,
   System.Globalization.CompareOptions options
);
public int Compare(
   System.String string1,
   int offset1,
   System.String string2,
   int offset2
);
public int Compare(
   System.String string1,
   int offset1,
   System.String string2,
   int offset2,
   System.Globalization.CompareOptions options
);
public int Compare(
   System.String string1,
   int offset1,
   int length1,
   System.String string2,
   int offset2,
   int length2
);
public int Compare(
   System.String string1,
   int offset1,
   int length1,
   System.String string2,
   int offset2,
   int length2,
   System.Globalization.CompareOptions options
);
public boolean Equals(
   System.Object value
);
public static System.Globalization.CompareInfo GetCompareInfo(
   int culture
```

```
);
public static System.Globalization.CompareInfo GetCompareInfo(
   System.String name
);
public static System.Globalization.CompareInfo GetCompareInfo(
   int culture,
   System.Reflection.Assembly assembly
);
public static System.Globalization.CompareInfo GetCompareInfo(
   System.String name,
   System.Reflection.Assembly assembly
);
public int GetHashCode();
public System.Globalization.SortKey GetSortKey(
   System.String source
);
public System.Globalization.SortKey GetSortKey(
   System.String source,
   System.Globalization.CompareOptions options
);
public int IndexOf(
   System.String source,
   char value
);
public int IndexOf(
   System.String source,
   System.String value
);
public int IndexOf(
   System.String source,
   char value,
   int startIndex
);
public int IndexOf(
   System.String source,
   char value,
   System.Globalization.CompareOptions options
);
public int IndexOf(
   System.String source,
   System.String value,
   int startIndex
);
public int IndexOf(
   System.String source,
   System.String value,
   System.Globalization.CompareOptions options
);
public int IndexOf(
   System.String source,
   char value,
   int startIndex,
   int count
);
public int IndexOf(
   System.String source,
   char value,
   int startIndex,
   System.Globalization.CompareOptions options
);
public int IndexOf(
   System.String source,
   System.String value,
   int startIndex,
   int count
);
public int IndexOf(
   System.String source,
```

```
   System.String value,
   int startIndex,
   System.Globalization.CompareOptions options
);
public int IndexOf(
   System.String source,
   char value,
   int startIndex,
   int count,
   System.Globalization.CompareOptions options
);
public int IndexOf(
   System.String source,
   System.String value,
   int startIndex,
   int count,
   System.Globalization.CompareOptions options
);
public boolean IsPrefix(
   System.String source,
   System.String prefix
);
public boolean IsPrefix(
   System.String source,
   System.String prefix,
   System.Globalization.CompareOptions options
);
public boolean IsSuffix(
   System.String source,
   System.String suffix
);
public boolean IsSuffix(
   System.String source,
   System.String suffix,
   System.Globalization.CompareOptions options
);
public int LastIndexOf(
   System.String source,
   char value
);
public int LastIndexOf(
   System.String source,
   System.String value
);
public int LastIndexOf(
   System.String source,
   char value,
   int startIndex
);
public int LastIndexOf(
   System.String source,
   char value,
   System.Globalization.CompareOptions options
);
public int LastIndexOf(
   System.String source,
   System.String value,
   int startIndex
);
public int LastIndexOf(
   System.String source,
   System.String value,
   System.Globalization.CompareOptions options
);
public int LastIndexOf(
   System.String source,
   char value,
   int startIndex,
```

```
      int count
   );
   public int LastIndexOf(
      System.String source,
      char value,
      int startIndex,
      System.Globalization.CompareOptions options
   );
   public int LastIndexOf(
      System.String source,
      System.String value,
      int startIndex,
      int count
   );
   public int LastIndexOf(
      System.String source,
      System.String value,
      int startIndex,
      System.Globalization.CompareOptions options
   );
   public int LastIndexOf(
      System.String source,
      char value,
      int startIndex,
      int count,
      System.Globalization.CompareOptions options
   );
   public int LastIndexOf(
      System.String source,
      System.String value,
      int startIndex,
      int count,
      System.Globalization.CompareOptions options
   );
   public System.String ToString();
```

CompareOptions Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum CompareOptions
```

Members

IgnoreCase

IgnoreKanaType

IgnoreNonSpace

IgnoreSymbols

IgnoreWidth

None

Ordinal

StringSort

CultureInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class CultureInfo
   extends System.Object
   implements System.ICloneable, System.IFormatProvider
```

Public Constructors

```
public CultureInfo(
   int culture
);
public CultureInfo(
   System.String name
);
public CultureInfo(
   int culture,
   boolean useUserOverride
);
public CultureInfo(
   System.String name,
   boolean useUserOverride
);
```

Public Properties

```
public System.Globalization.Calendar get_Calendar();
public System.Globalization.CompareInfo get_CompareInfo();
public static System.Globalization.CultureInfo get_CurrentCulture();
public static System.Globalization.CultureInfo get_CurrentUICulture();
public System.Globalization.DateTimeFormatInfo get_DateTimeFormat();
public void set_DateTimeFormat(
   System.Globalization.DateTimeFormatInfo value
);
public System.String get_DisplayName();
public System.String get_EnglishName();
public static System.Globalization.CultureInfo get_InstalledUICulture();
public static System.Globalization.CultureInfo get_InvariantCulture();
public boolean get_IsNeutralCulture();
public boolean get_IsReadOnly();
public int get_LCID();
public System.String get_Name();
public System.String get_NativeName();
public System.Globalization.NumberFormatInfo get_NumberFormat();
public void set_NumberFormat(
   System.Globalization.NumberFormatInfo value
);
public System.Globalization.Calendar[] get_OptionalCalendars();
public System.Globalization.CultureInfo get_Parent();
public System.Globalization.TextInfo get_TextInfo();
public System.String get_ThreeLetterISOLanguageName();
public System.String get_ThreeLetterWindowsLanguageName();
public System.String get_TwoLetterISOLanguageName();
public boolean get_UseUserOverride();
```

Public Methods

```
public void ClearCachedData();
public System.Object Clone();
public static System.Globalization.CultureInfo CreateSpecificCulture(
   System.String name
);
public boolean Equals(
   System.Object value
);
public static System.Globalization.CultureInfo[] GetCultures(
   System.Globalization.CultureTypes types
);
public System.Object GetFormat(
   System.Type formatType
);
public int GetHashCode();
public static System.Globalization.CultureInfo ReadOnly(
   System.Globalization.CultureInfo ci
);
public System.String ToString();
```

CultureTypes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum CultureTypes
```

Members

AllCultures

InstalledWin32Cultures

NeutralCultures

SpecificCultures

DateTimeFormatInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DateTimeFormatInfo
   extends System.Object
   implements System.ICloneable, System.IFormatProvider
```

Public Constructors

```
public DateTimeFormatInfo();
```

Public Properties

```
public System.String[] get_AbbreviatedDayNames();
public void set_AbbreviatedDayNames(
   System.String[] value
);
public System.String[] get_AbbreviatedMonthNames();
public void set_AbbreviatedMonthNames(
   System.String[] value
);
public System.String get_AMDesignator();
public void set_AMDesignator(
   System.String value
);
public System.Globalization.Calendar get_Calendar();
public void set_Calendar(
   System.Globalization.Calendar value
);
public System.Globalization.CalendarWeekRule get_CalendarWeekRule();
public void set_CalendarWeekRule(
   System.Globalization.CalendarWeekRule value
);
public static System.Globalization.DateTimeFormatInfo get_CurrentInfo();
public System.String get_DateSeparator();
public void set_DateSeparator(
   System.String value
);
public System.String[] get_DayNames();
public void set_DayNames(
   System.String[] value
);
public System.DayOfWeek get_FirstDayOfWeek();
public void set_FirstDayOfWeek(
   System.DayOfWeek value
);
public System.String get_FullDateTimePattern();
public void set_FullDateTimePattern(
   System.String value
);
public static System.Globalization.DateTimeFormatInfo get_InvariantInfo();
public boolean get_IsReadOnly();
public System.String get_LongDatePattern();
public void set_LongDatePattern(
   System.String value
);
public System.String get_LongTimePattern();
```

```
public void set_LongTimePattern(
   System.String value
);
public System.String get_MonthDayPattern();
public void set_MonthDayPattern(
   System.String value
);
public System.String[] get_MonthNames();
public void set_MonthNames(
   System.String[] value
);
public System.String get_PMDesignator();
public void set_PMDesignator(
   System.String value
);
public System.String get_RFC1123Pattern();
public System.String get_ShortDatePattern();
public void set_ShortDatePattern(
   System.String value
);
public System.String get_ShortTimePattern();
public void set_ShortTimePattern(
   System.String value
);
public System.String get_SortableDateTimePattern();
public System.String get_TimeSeparator();
public void set_TimeSeparator(
   System.String value
);
public System.String get_UniversalSortableDateTimePattern();
public System.String get_YearMonthPattern();
public void set_YearMonthPattern(
   System.String value
);
```

Public Methods

```
public final System.Object Clone();
public System.String GetAbbreviatedDayName(
   System.DayOfWeek dayofweek
);
public System.String GetAbbreviatedEraName(
   int era
);
public System.String GetAbbreviatedMonthName(
   int month
);
public System.String[] GetAllDateTimePatterns();
public System.String[] GetAllDateTimePatterns(
   char format
);
public System.String GetDayName(
   System.DayOfWeek dayofweek
);
public int GetEra(
   System.String eraName
);
public System.String GetEraName(
   int era
);
public final System.Object GetFormat(
   System.Type formatType
);
public static System.Globalization.DateTimeFormatInfo GetInstance(
   System.IFormatProvider provider
);
```

```
public System.String GetMonthName(
   int month
);
public static System.Globalization.DateTimeFormatInfo ReadOnly(
   System.Globalization.DateTimeFormatInfo dtfi
);
```

DateTimeStyles Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum DateTimeStyles
```

Members

AdjustToUniversal

AllowInnerWhite

AllowLeadingWhite

AllowTrailingWhite

AllowWhiteSpaces

NoCurrentDateDefault

None

DaylightTime Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DaylightTime
   extends System.Object
```

Public Constructors

```
public DaylightTime(
   System.DateTime start,
   System.DateTime end,
   System.TimeSpan delta
);
```

Public Properties

```
public System.TimeSpan get_Delta();
public System.DateTime get_End();
public System.DateTime get_Start();
```

GregorianCalendar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class GregorianCalendar
   extends System.Globalization.Calendar
```

Public Constructors

```
public GregorianCalendar();
public GregorianCalendar(
   System.Globalization.GregorianCalendarTypes type
);
```

Public Properties

```
public System.Globalization.GregorianCalendarTypes get_CalendarType();
public void set_CalendarType(
   System.Globalization.GregorianCalendarTypes value
);
public int[] get_Eras();
public int get_TwoDigitYearMax();
public void set_TwoDigitYearMax(
   int value
);
```

Public Methods

```
public System.DateTime AddMonths(
   System.DateTime time,
   int months
);
public System.DateTime AddWeeks(
   System.DateTime time,
   int weeks
);
public System.DateTime AddYears(
   System.DateTime time,
   int years
);
public int GetDayOfMonth(
   System.DateTime time
);
public System.DayOfWeek GetDayOfWeek(
   System.DateTime time
);
public int GetDayOfYear(
   System.DateTime time
);
public int GetDaysInMonth(
   int year,
   int month,
   int era
);
public int GetDaysInYear(
   int year,
```

```
      int era
   );
   public int GetEra(
      System.DateTime time
   );
   public int GetMonth(
      System.DateTime time
   );
   public int GetMonthsInYear(
      int year,
      int era
   );
   public int GetYear(
      System.DateTime time
   );
   public boolean IsLeapDay(
      int year,
      int month,
      int day,
      int era
   );
   public boolean IsLeapMonth(
      int year,
      int month,
      int era
   );
   public boolean IsLeapYear(
      int year,
      int era
   );
   public System.DateTime ToDateTime(
      int year,
      int month,
      int day,
      int hour,
      int minute,
      int second,
      int millisecond,
      int era
   );
   public int ToFourDigitYear(
      int year
   );
```

Public Fields

```
   public static int ADEra;
```

GregorianCalendarTypes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum GregorianCalendarTypes
```

Members

Arabic

Localized

MiddleEastFrench

TransliteratedEnglish

TransliteratedFrench

USEnglish

HebrewCalendar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class HebrewCalendar
   extends System.Globalization.Calendar
```

Public Constructors

```
public HebrewCalendar();
```

Public Properties

```
public int[] get_Eras();
public int get_TwoDigitYearMax();
public void set_TwoDigitYearMax(
   int value
);
```

Public Methods

```
public System.DateTime AddMonths(
   System.DateTime time,
   int months
);
public System.DateTime AddYears(
   System.DateTime time,
   int years
);
public int GetDayOfMonth(
   System.DateTime time
);
public System.DayOfWeek GetDayOfWeek(
   System.DateTime time
);
public int GetDayOfYear(
   System.DateTime time
);
public int GetDaysInMonth(
   int year,
   int month,
   int era
);
public int GetDaysInYear(
   int year,
   int era
);
public int GetEra(
   System.DateTime time
);
public int GetMonth(
   System.DateTime time
);
public int GetMonthsInYear(
   int year,
   int era
```

```
);
public int GetYear(
   System.DateTime time
);
public boolean IsLeapDay(
   int year,
   int month,
   int day,
   int era
);
public boolean IsLeapMonth(
   int year,
   int month,
   int era
);
public boolean IsLeapYear(
   int year,
   int era
);
public System.DateTime ToDateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second,
   int millisecond,
   int era
);
public int ToFourDigitYear(
   int year
);
```

Public Fields

```
public static final int HebrewEra;
```

HijriCalendar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class HijriCalendar
   extends System.Globalization.Calendar
```

Public Constructors

```
public HijriCalendar();
```

Public Properties

```
public int[] get_Eras();
public int get_TwoDigitYearMax();
public void set_TwoDigitYearMax(
   int value
);
```

Public Methods

```
public System.DateTime AddMonths(
   System.DateTime time,
   int months
);
public System.DateTime AddYears(
   System.DateTime time,
   int years
);
public int GetDayOfMonth(
   System.DateTime time
);
public System.DayOfWeek GetDayOfWeek(
   System.DateTime time
);
public int GetDayOfYear(
   System.DateTime time
);
public int GetDaysInMonth(
   int year,
   int month,
   int era
);
public int GetDaysInYear(
   int year,
   int era
);
public int GetEra(
   System.DateTime time
);
public int GetMonth(
   System.DateTime time
);
public int GetMonthsInYear(
   int year,
   int era
```

```
);
public int GetYear(
   System.DateTime time
);
public boolean IsLeapDay(
   int year,
   int month,
   int day,
   int era
);
public boolean IsLeapMonth(
   int year,
   int month,
   int era
);
public boolean IsLeapYear(
   int year,
   int era
);
public System.DateTime ToDateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second,
   int millisecond,
   int era
);
public int ToFourDigitYear(
   int year
);
```

Public Fields

```
public static final int HijriEra;
```

JapaneseCalendar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class JapaneseCalendar
   extends System.Globalization.Calendar
```

Public Constructors

```
public JapaneseCalendar();
```

Public Properties

```
public int[] get_Eras();
public int get_TwoDigitYearMax();
public void set_TwoDigitYearMax(
   int value
);
```

Public Methods

```
public System.DateTime AddMonths(
   System.DateTime time,
   int months
);
public System.DateTime AddYears(
   System.DateTime time,
   int years
);
public int GetDayOfMonth(
   System.DateTime time
);
public System.DayOfWeek GetDayOfWeek(
   System.DateTime time
);
public int GetDayOfYear(
   System.DateTime time
);
public int GetDaysInMonth(
   int year,
   int month,
   int era
);
public int GetDaysInYear(
   int year,
   int era
);
public int GetEra(
   System.DateTime time
);
public int GetMonth(
   System.DateTime time
);
public int GetMonthsInYear(
   int year,
   int era
```

```
);
public int GetYear(
   System.DateTime time
);
public boolean IsLeapDay(
   int year,
   int month,
   int day,
   int era
);
public boolean IsLeapMonth(
   int year,
   int month,
   int era
);
public boolean IsLeapYear(
   int year,
   int era
);
public System.DateTime ToDateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second,
   int millisecond,
   int era
);
public int ToFourDigitYear(
   int year
);
```

JulianCalendar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class JulianCalendar
   extends System.Globalization.Calendar
```

Public Constructors

```
public JulianCalendar();
```

Public Properties

```
public int[] get_Eras();
public int get_TwoDigitYearMax();
public void set_TwoDigitYearMax(
   int value
);
```

Public Methods

```
public System.DateTime AddMonths(
   System.DateTime time,
   int months
);
public System.DateTime AddYears(
   System.DateTime time,
   int years
);
public int GetDayOfMonth(
   System.DateTime time
);
public System.DayOfWeek GetDayOfWeek(
   System.DateTime time
);
public int GetDayOfYear(
   System.DateTime time
);
public int GetDaysInMonth(
   int year,
   int month,
   int era
);
public int GetDaysInYear(
   int year,
   int era
);
public int GetEra(
   System.DateTime time
);
public int GetMonth(
   System.DateTime time
);
public int GetMonthsInYear(
   int year,
   int era
```

```
);
public int GetYear(
   System.DateTime time
);
public boolean IsLeapDay(
   int year,
   int month,
   int day,
   int era
);
public boolean IsLeapMonth(
   int year,
   int month,
   int era
);
public boolean IsLeapYear(
   int year,
   int era
);
public System.DateTime ToDateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second,
   int millisecond,
   int era
);
public int ToFourDigitYear(
   int year
);
```

Public Fields

```
public static final int JulianEra;
```

KoreanCalendar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class KoreanCalendar
   extends System.Globalization.Calendar
```

Public Constructors

```
public KoreanCalendar();
```

Public Properties

```
public int[] get_Eras();
public int get_TwoDigitYearMax();
public void set_TwoDigitYearMax(
   int value
);
```

Public Methods

```
public System.DateTime AddMonths(
   System.DateTime time,
   int months
);
public System.DateTime AddYears(
   System.DateTime time,
   int years
);
public int GetDayOfMonth(
   System.DateTime time
);
public System.DayOfWeek GetDayOfWeek(
   System.DateTime time
);
public int GetDayOfYear(
   System.DateTime time
);
public int GetDaysInMonth(
   int year,
   int month,
   int era
);
public int GetDaysInYear(
   int year,
   int era
);
public int GetEra(
   System.DateTime time
);
public int GetMonth(
   System.DateTime time
);
public int GetMonthsInYear(
   int year,
   int era
```

```
);
public int GetYear(
   System.DateTime time
);
public boolean IsLeapDay(
   int year,
   int month,
   int day,
   int era
);
public boolean IsLeapMonth(
   int year,
   int month,
   int era
);
public boolean IsLeapYear(
   int year,
   int era
);
public System.DateTime ToDateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second,
   int millisecond,
   int era
);
public int ToFourDigitYear(
   int year
);
```

Public Fields

```
public static int KoreanEra;
```

NumberFormatInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class NumberFormatInfo
   extends System.Object
   implements System.ICloneable, System.IFormatProvider
```

Public Constructors

```
public NumberFormatInfo();
```

Public Properties

```
public int get_CurrencyDecimalDigits();
public void set_CurrencyDecimalDigits(
   int value
);
public System.String get_CurrencyDecimalSeparator();
public void set_CurrencyDecimalSeparator(
   System.String value
);
public System.String get_CurrencyGroupSeparator();
public void set_CurrencyGroupSeparator(
   System.String value
);
public int[] get_CurrencyGroupSizes();
public void set_CurrencyGroupSizes(
   int[] value
);
public int get_CurrencyNegativePattern();
public void set_CurrencyNegativePattern(
   int value
);
public int get_CurrencyPositivePattern();
public void set_CurrencyPositivePattern(
   int value
);
public System.String get_CurrencySymbol();
public void set_CurrencySymbol(
   System.String value
);
public static System.Globalization.NumberFormatInfo get_CurrentInfo();
public static System.Globalization.NumberFormatInfo get_InvariantInfo();
public boolean get_IsReadOnly();
public System.String get_NaNSymbol();
public void set_NaNSymbol(
   System.String value
);
public System.String get_NegativeInfinitySymbol();
public void set_NegativeInfinitySymbol(
   System.String value
);
public System.String get_NegativeSign();
public void set_NegativeSign(
   System.String value
);
public int get_NumberDecimalDigits();
```

```
public void set_NumberDecimalDigits(
   int value
);
public System.String get_NumberDecimalSeparator();
public void set_NumberDecimalSeparator(
   System.String value
);
public System.String get_NumberGroupSeparator();
public void set_NumberGroupSeparator(
   System.String value
);
public int[] get_NumberGroupSizes();
public void set_NumberGroupSizes(
   int[] value
);
public int get_NumberNegativePattern();
public void set_NumberNegativePattern(
   int value
);
public int get_PercentDecimalDigits();
public void set_PercentDecimalDigits(
   int value
);
public System.String get_PercentDecimalSeparator();
public void set_PercentDecimalSeparator(
   System.String value
);
public System.String get_PercentGroupSeparator();
public void set_PercentGroupSeparator(
   System.String value
);
public int[] get_PercentGroupSizes();
public void set_PercentGroupSizes(
   int[] value
);
public int get_PercentNegativePattern();
public void set_PercentNegativePattern(
   int value
);
public int get_PercentPositivePattern();
public void set_PercentPositivePattern(
   int value
);
public System.String get_PercentSymbol();
public void set_PercentSymbol(
   System.String value
);
public System.String get_PerMilleSymbol();
public void set_PerMilleSymbol(
   System.String value
);
public System.String get_PositiveInfinitySymbol();
public void set_PositiveInfinitySymbol(
   System.String value
);
public System.String get_PositiveSign();
public void set_PositiveSign(
   System.String value
);
```

Public Methods

```
public final System.Object Clone();
public final System.Object GetFormat(
   System.Type formatType
);
```

```
public static System.Globalization.NumberFormatInfo GetInstance(
   System.IFormatProvider formatProvider
);
public static System.Globalization.NumberFormatInfo ReadOnly(
   System.Globalization.NumberFormatInfo nfi
);
```

NumberStyles Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum NumberStyles
```

Members

AllowCurrencySymbol

AllowDecimalPoint

AllowExponent

AllowHexSpecifier

AllowLeadingSign

AllowLeadingWhite

AllowParentheses

AllowThousands

AllowTrailingSign

AllowTrailingWhite

Any

Currency

Float

HexNumber

Integer

None

Number

RegionInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RegionInfo
   extends System.Object
```

Public Constructors

```
public RegionInfo(
   int culture
);
public RegionInfo(
   System.String name
);
```

Public Properties

```
public System.String get_CurrencySymbol();
public static System.Globalization.RegionInfo get_CurrentRegion();
public System.String get_DisplayName();
public System.String get_EnglishName();
public boolean get_IsMetric();
public System.String get_ISOCurrencySymbol();
public System.String get_Name();
public System.String get_ThreeLetterISORegionName();
public System.String get_ThreeLetterWindowsRegionName();
public System.String get_TwoLetterISORegionName();
```

Public Methods

```
public boolean Equals(
   System.Object value
);
public int GetHashCode();
public System.String ToString();
```

SortKey Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SortKey
   extends System.Object
```

Public Properties

```
public ubyte[] get_KeyData();
public System.String get_OriginalString();
```

Public Methods

```
public static int Compare(
   System.Globalization.SortKey sortkey1,
   System.Globalization.SortKey sortkey2
);
public boolean Equals(
   System.Object value
);
public int GetHashCode();
public System.String ToString();
```

StringInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StringInfo
   extends System.Object
```

Public Constructors

```
public StringInfo();
```

Public Methods

```
public static System.String GetNextTextElement(
   System.String str
);
public static System.String GetNextTextElement(
   System.String str,
   int index
);
public static System.Globalization.TextElementEnumerator GetTextElementEnumerator(
   System.String str
);
public static System.Globalization.TextElementEnumerator GetTextElementEnumerator(
   System.String str,
   int index
);
public static int[] ParseCombiningCharacters(
   System.String str
);
```

TaiwanCalendar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class TaiwanCalendar
   extends System.Globalization.Calendar
```

Public Constructors

```
public TaiwanCalendar();
```

Public Properties

```
public int[] get_Eras();
public int get_TwoDigitYearMax();
public void set_TwoDigitYearMax(
   int value
);
```

Public Methods

```
public System.DateTime AddMonths(
   System.DateTime time,
   int months
);
public System.DateTime AddYears(
   System.DateTime time,
   int years
);
public int GetDayOfMonth(
   System.DateTime time
);
public System.DayOfWeek GetDayOfWeek(
   System.DateTime time
);
public int GetDayOfYear(
   System.DateTime time
);
public int GetDaysInMonth(
   int year,
   int month,
   int era
);
public int GetDaysInYear(
   int year,
   int era
);
public int GetEra(
   System.DateTime time
);
public int GetMonth(
   System.DateTime time
);
public int GetMonthsInYear(
   int year,
   int era
```

```
);
public int GetYear(
   System.DateTime time
);
public boolean IsLeapDay(
   int year,
   int month,
   int day,
   int era
);
public boolean IsLeapMonth(
   int year,
   int month,
   int era
);
public boolean IsLeapYear(
   int year,
   int era
);
public System.DateTime ToDateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second,
   int millisecond,
   int era
);
public int ToFourDigitYear(
   int year
);
```

TextElementEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class TextElementEnumerator
   extends System.Object
   implements System.Collections.IEnumerator
```

Public Properties

```
public final System.Object get_Current();
public int get_ElementIndex();
```

Public Methods

```
public System.String GetTextElement();
public final boolean MoveNext();
public final void Reset();
```

TextInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class TextInfo
   extends System.Object
   implements System.Runtime.Serialization.IDeserializationCallback
```

Public Properties

```
public int get_ANSICodePage();
public int get_EBCDICCodePage();
public System.String get_ListSeparator();
public int get_MacCodePage();
public int get_OEMCodePage();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public char ToLower(
   char c
);
public System.String ToLower(
   System.String str
);
public System.String ToString();
public System.String ToTitleCase(
   System.String str
);
public char ToUpper(
   char c
);
public System.String ToUpper(
   System.String str
);
```

ThaiBuddhistCalendar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ThaiBuddhistCalendar
   extends System.Globalization.Calendar
```

Public Constructors

```
public ThaiBuddhistCalendar();
```

Public Properties

```
public int[] get_Eras();
public int get_TwoDigitYearMax();
public void set_TwoDigitYearMax(
   int value
);
```

Public Methods

```
public System.DateTime AddMonths(
   System.DateTime time,
   int months
);
public System.DateTime AddYears(
   System.DateTime time,
   int years
);
public int GetDayOfMonth(
   System.DateTime time
);
public System.DayOfWeek GetDayOfWeek(
   System.DateTime time
);
public int GetDayOfYear(
   System.DateTime time
);
public int GetDaysInMonth(
   int year,
   int month,
   int era
);
public int GetDaysInYear(
   int year,
   int era
);
public int GetEra(
   System.DateTime time
);
public int GetMonth(
   System.DateTime time
);
public int GetMonthsInYear(
   int year,
   int era
```

```
);
public int GetYear(
   System.DateTime time
);
public boolean IsLeapDay(
   int year,
   int month,
   int day,
   int era
);
public boolean IsLeapMonth(
   int year,
   int month,
   int era
);
public boolean IsLeapYear(
   int year,
   int era
);
public System.DateTime ToDateTime(
   int year,
   int month,
   int day,
   int hour,
   int minute,
   int second,
   int millisecond,
   int era
);
public int ToFourDigitYear(
   int year
);
```

Public Fields

```
public static int ThaiBuddhistEra;
```

UnicodeCategory Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum UnicodeCategory
```

Members

ClosePunctuation

ConnectorPunctuation

Control

CurrencySymbol

DashPunctuation

DecimalDigitNumber

EnclosingMark

FinalQuotePunctuation

Format

InitialQuotePunctuation

LetterNumber

LineSeparator

LowercaseLetter

MathSymbol

ModifierLetter

ModifierSymbol

NonSpacingMark

OpenPunctuation

OtherLetter

OtherNotAssigned

OtherNumber

OtherPunctuation

OtherSymbol

ParagraphSeparator

PrivateUse

SpaceSeparator

SpacingCombiningMark

Surrogate

TitlecaseLetter

UppercaseLetter

System.IO Namespace (Visual J# Syntax)

Classes

BinaryReader

BinaryWriter

BufferedStream

Directory

DirectoryInfo

DirectoryNotFoundException

EndOfStreamException

ErrorEventArgs

File

FileInfo

FileLoadException

FileNotFoundException

FileStream

FileSystemEventArgs

FileSystemInfo

FileSystemWatcher

InternalBufferOverflowException

IODescriptionAttribute

IOException

MemoryStream

Path

PathTooLongException

RenamedEventArgs

Stream

StreamReader

StreamWriter

StringReader

StringWriter

TextReader

TextWriter

Structures

WaitForChangedResult

Delegates

ErrorEventHandler

FileSystemEventHandler

RenamedEventHandler

Enumerations

FileAccess

FileAttributes

FileMode

FileShare

NotifyFilters

SeekOrigin

WatcherChangeTypes

BinaryReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BinaryReader
   extends System.Object
   implements System.IDisposable
```

Public Constructors

```
public BinaryReader(
   System.IO.Stream input
);
public BinaryReader(
   System.IO.Stream input,
   System.Text.Encoding encoding
);
```

Public Properties

```
public System.IO.Stream get_BaseStream();
```

Public Methods

```
public void Close();
public int PeekChar();
public int Read();
public int Read(
   ubyte[] buffer,
   int index,
   int count
);
public int Read(
   char[] buffer,
   int index,
   int count
);
public boolean ReadBoolean();
public ubyte ReadByte();
public ubyte[] ReadBytes(
   int count
);
public char ReadChar();
public char[] ReadChars(
   int count
);
public System.Decimal ReadDecimal();
public double ReadDouble();
public short ReadInt16();
public int ReadInt32();
public long ReadInt64();
public float ReadSingle();
public System.String ReadString();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void FillBuffer(
   int numBytes
);
protected int Read7BitEncodedInt();
```

BinaryWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class BinaryWriter
   extends System.Object
   implements System.IDisposable
```

Public Constructors

```
public BinaryWriter(
   System.IO.Stream output
);
public BinaryWriter(
   System.IO.Stream output,
   System.Text.Encoding encoding
);
```

Public Properties

```
public System.IO.Stream get_BaseStream();
```

Public Methods

```
public void Close();
public void Flush();
public long Seek(
   int offset,
   System.IO.SeekOrigin origin
);
public void Write(
   boolean value
);
public void Write(
   ubyte value
);
public void Write(
   ubyte[] buffer
);
public void Write(
   char ch
);
public void Write(
   char[] chars
);
public void Write(
   double value
);
public void Write(
   short value
);
public void Write(
   int value
);
public void Write(
   long value
```

```
);
public void Write(
   float value
);
public void Write(
   System.Decimal value
);
public void Write(
   System.String value
);
public void Write(
   ubyte[] buffer,
   int index,
   int count
);
public void Write(
   char[] chars,
   int index,
   int count
);
```

Public Fields

```
public static final System.IO.BinaryWriter Null;
```

Protected Constructors

```
protected BinaryWriter();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Write7BitEncodedInt(
   int value
);
```

Protected Fields

```
protected System.IO.Stream OutStream;
```

BufferedStream Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BufferedStream
   extends System.IO.Stream
```

Public Constructors

```
public BufferedStream(
   System.IO.Stream stream
);
public BufferedStream(
   System.IO.Stream stream,
   int bufferSize
);
```

Public Properties

```
public boolean get_CanRead();
public boolean get_CanSeek();
public boolean get_CanWrite();
public long get_Length();
public long get_Position();
public void set_Position(
   long value
);
```

Public Methods

```
public void Close();
public void Flush();
public int Read(
   ubyte[] array,
   int offset,
   int count
);
public int ReadByte();
public long Seek(
   long offset,
   System.IO.SeekOrigin origin
);
public void SetLength(
   long value
);
public void Write(
   ubyte[] array,
   int offset,
   int count
);
public void WriteByte(
   ubyte value
);
```

Directory Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Directory
   extends System.Object
```

Public Methods

```
public static System.IO.DirectoryInfo CreateDirectory(
   System.String path
);
public static void Delete(
   System.String path
);
public static void Delete(
   System.String path,
   boolean recursive
);
public static boolean Exists(
   System.String path
);
public static System.DateTime GetCreationTime(
   System.String path
);
public static System.String GetCurrentDirectory();
public static System.String[] GetDirectories(
   System.String path
);
public static System.String[] GetDirectories(
   System.String path,
   System.String searchPattern
);
public static System.String GetDirectoryRoot(
   System.String path
);
public static System.String[] GetFiles(
   System.String path
);
public static System.String[] GetFiles(
   System.String path,
   System.String searchPattern
);
public static System.String[] GetFileSystemEntries(
   System.String path
);
public static System.String[] GetFileSystemEntries(
   System.String path,
   System.String searchPattern
);
public static System.DateTime GetLastAccessTime(
   System.String path
);
public static System.DateTime GetLastWriteTime(
   System.String path
);
public static System.String[] GetLogicalDrives();
public static System.IO.DirectoryInfo GetParent(
   System.String path
);
public static void Move(
```

```
   System.String sourceDirName,
   System.String destDirName
);
public static void SetCreationTime(
   System.String path,
   System.DateTime creationTime
);
public static void SetCurrentDirectory(
   System.String path
);
public static void SetLastAccessTime(
   System.String path,
   System.DateTime lastAccessTime
);
public static void SetLastWriteTime(
   System.String path,
   System.DateTime lastWriteTime
);
```

DirectoryInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DirectoryInfo
   extends System.IO.FileSystemInfo
```

Public Constructors

```
public DirectoryInfo(
   System.String path
);
```

Public Properties

```
public boolean get_Exists();
public System.String get_Name();
public System.IO.DirectoryInfo get_Parent();
public System.IO.DirectoryInfo get_Root();
```

Public Methods

```
public void Create();
public System.IO.DirectoryInfo CreateSubdirectory(
   System.String path
);
public void Delete();
public void Delete(
   boolean recursive
);
public System.IO.DirectoryInfo[] GetDirectories();
public System.IO.DirectoryInfo[] GetDirectories(
   System.String searchPattern
);
public System.IO.FileInfo[] GetFiles();
public System.IO.FileInfo[] GetFiles(
   System.String searchPattern
);
public System.IO.FileSystemInfo[] GetFileSystemInfos();
public System.IO.FileSystemInfo[] GetFileSystemInfos(
   System.String searchPattern
);
public void MoveTo(
   System.String destDirName
);
public System.String ToString();
```

DirectoryNotFoundException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DirectoryNotFoundException
   extends System.IO.IOException
```

Public Constructors

```
public DirectoryNotFoundException();
public DirectoryNotFoundException(
   System.String message
);
public DirectoryNotFoundException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected DirectoryNotFoundException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

EndOfStreamException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class EndOfStreamException
   extends System.IO.IOException
```

Public Constructors

```
public EndOfStreamException();
public EndOfStreamException(
   System.String message
);
public EndOfStreamException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected EndOfStreamException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ErrorEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ErrorEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ErrorEventArgs(
   System.Exception exception
);
```

Public Methods

```
public System.Exception GetException();
```

ErrorEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ErrorEventHandler(
   System.Object sender,
   System.IO.ErrorEventArgs e
);
```

File Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class File
   extends System.Object
```

Public Methods

```
public static System.IO.StreamWriter AppendText(
   System.String path
);
public static void Copy(
   System.String sourceFileName,
   System.String destFileName
);
public static void Copy(
   System.String sourceFileName,
   System.String destFileName,
   boolean overwrite
);
public static System.IO.FileStream Create(
   System.String path
);
public static System.IO.FileStream Create(
   System.String path,
   int bufferSize
);
public static System.IO.StreamWriter CreateText(
   System.String path
);
public static void Delete(
   System.String path
);
public static boolean Exists(
   System.String path
);
public static System.IO.FileAttributes GetAttributes(
   System.String path
);
public static System.DateTime GetCreationTime(
   System.String path
);
public static System.DateTime GetLastAccessTime(
   System.String path
);
public static System.DateTime GetLastWriteTime(
   System.String path
);
public static void Move(
   System.String sourceFileName,
   System.String destFileName
);
public static System.IO.FileStream Open(
   System.String path,
   System.IO.FileMode mode
);
public static System.IO.FileStream Open(
   System.String path,
   System.IO.FileMode mode,
   System.IO.FileAccess access
```

```
);
public static System.IO.FileStream Open(
   System.String path,
   System.IO.FileMode mode,
   System.IO.FileAccess access,
   System.IO.FileShare share
);
public static System.IO.FileStream OpenRead(
   System.String path
);
public static System.IO.StreamReader OpenText(
   System.String path
);
public static System.IO.FileStream OpenWrite(
   System.String path
);
public static void SetAttributes(
   System.String path,
   System.IO.FileAttributes fileAttributes
);
public static void SetCreationTime(
   System.String path,
   System.DateTime creationTime
);
public static void SetLastAccessTime(
   System.String path,
   System.DateTime lastAccessTime
);
public static void SetLastWriteTime(
   System.String path,
   System.DateTime lastWriteTime
);
```

FileAccess Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum FileAccess
```

Members

Read

ReadWrite

Write

FileAttributes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum FileAttributes
```

Members

Archive

Compressed

Device

Directory

Encrypted

Hidden

Normal

NotContentIndexed

Offline

ReadOnly

ReparsePoint

SparseFile

System

Temporary

FileInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class FileInfo
   extends System.IO.FileSystemInfo
```

Public Constructors

```
public FileInfo(
   System.String fileName
);
```

Public Properties

```
public System.IO.DirectoryInfo get_Directory();
public System.String get_DirectoryName();
public boolean get_Exists();
public long get_Length();
public System.String get_Name();
```

Public Methods

```
public System.IO.StreamWriter AppendText();
public System.IO.FileInfo CopyTo(
   System.String destFileName
);
public System.IO.FileInfo CopyTo(
   System.String destFileName,
   boolean overwrite
);
public System.IO.FileStream Create();
public System.IO.StreamWriter CreateText();
public void Delete();
public void MoveTo(
   System.String destFileName
);
public System.IO.FileStream Open(
   System.IO.FileMode mode
);
public System.IO.FileStream Open(
   System.IO.FileMode mode,
   System.IO.FileAccess access
);
public System.IO.FileStream Open(
   System.IO.FileMode mode,
   System.IO.FileAccess access,
   System.IO.FileShare share
);
public System.IO.FileStream OpenRead();
public System.IO.StreamReader OpenText();
public System.IO.FileStream OpenWrite();
public System.String ToString();
```

FileLoadException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class FileLoadException
   extends System.IO.IOException
```

Public Constructors

```
public FileLoadException();
public FileLoadException(
   System.String message
);
public FileLoadException(
   System.String message,
   System.Exception inner
);
public FileLoadException(
   System.String message,
   System.String fileName
);
public FileLoadException(
   System.String message,
   System.String fileName,
   System.Exception inner
);
```

Public Properties

```
public System.String get_FileName();
public System.String get_FusionLog();
public System.String get_Message();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public System.String ToString();
```

Protected Constructors

```
protected FileLoadException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

FileMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum FileMode
```

Members

Append

Create

CreateNew

Open

OpenOrCreate

Truncate

FileNotFoundException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class FileNotFoundException
   extends System.IO.IOException
```

Public Constructors

```
public FileNotFoundException();
public FileNotFoundException(
   System.String message
);
public FileNotFoundException(
   System.String message,
   System.Exception innerException
);
public FileNotFoundException(
   System.String message,
   System.String fileName
);
public FileNotFoundException(
   System.String message,
   System.String fileName,
   System.Exception innerException
);
```

Public Properties

```
public System.String get_FileName();
public System.String get_FusionLog();
public System.String get_Message();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public System.String ToString();
```

Protected Constructors

```
protected FileNotFoundException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

FileShare Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum FileShare
```

Members

Inheritable

None

Read

ReadWrite

Write

FileStream Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FileStream
   extends System.IO.Stream
```

Public Constructors

```
public FileStream(
   System.IntPtr handle,
   System.IO.FileAccess access
);
public FileStream(
   System.String path,
   System.IO.FileMode mode
);
public FileStream(
   System.IntPtr handle,
   System.IO.FileAccess access,
   boolean ownsHandle
);
public FileStream(
   System.String path,
   System.IO.FileMode mode,
   System.IO.FileAccess access
);
public FileStream(
   System.IntPtr handle,
   System.IO.FileAccess access,
   boolean ownsHandle,
   int bufferSize
);
public FileStream(
   System.String path,
   System.IO.FileMode mode,
   System.IO.FileAccess access,
   System.IO.FileShare share
);
public FileStream(
   System.IntPtr handle,
   System.IO.FileAccess access,
   boolean ownsHandle,
   int bufferSize,
   boolean isAsync
);
public FileStream(
   System.String path,
   System.IO.FileMode mode,
   System.IO.FileAccess access,
   System.IO.FileShare share,
   int bufferSize
);
public FileStream(
   System.String path,
   System.IO.FileMode mode,
   System.IO.FileAccess access,
   System.IO.FileShare share,
   int bufferSize,
   boolean useAsync
);
```

Public Properties

```
public boolean get_CanRead();
public boolean get_CanSeek();
public boolean get_CanWrite();
public System.IntPtr get_Handle();
public boolean get_IsAsync();
public long get_Length();
public System.String get_Name();
public long get_Position();
public void set_Position(
   long value
);
```

Public Methods

```
public System.IAsyncResult BeginRead(
   ubyte[] array,
   int offset,
   int numBytes,
   System.AsyncCallback userCallback,
   System.Object stateObject
);
public System.IAsyncResult BeginWrite(
   ubyte[] array,
   int offset,
   int numBytes,
   System.AsyncCallback userCallback,
   System.Object stateObject
);
public void Close();
public int EndRead(
   System.IAsyncResult asyncResult
);
public void EndWrite(
   System.IAsyncResult asyncResult
);
public void Flush();
public void Lock(
   long position,
   long length
);
public int Read(
   ubyte[] array,
   int offset,
   int count
);
public int ReadByte();
public long Seek(
   long offset,
   System.IO.SeekOrigin origin
);
public void SetLength(
   long value
);
public void Unlock(
   long position,
   long length
);
public void Write(
   ubyte[] array,
   int offset,
   int count
```

```
);
public void WriteByte(
   ubyte value
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

FileSystemEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FileSystemEventArgs
   extends System.EventArgs
```

Public Constructors

```
public FileSystemEventArgs(
   System.IO.WatcherChangeTypes changeType,
   System.String directory,
   System.String name
);
```

Public Properties

```
public System.IO.WatcherChangeTypes get_ChangeType();
public System.String get_FullPath();
public System.String get_Name();
```

FileSystemEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void FileSystemEventHandler(
   System.Object sender,
   System.IO.FileSystemEventArgs e
);
```

FileSystemInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class FileSystemInfo
   extends System.MarshalByRefObject
```

Public Properties

```
public System.IO.FileAttributes get_Attributes();
public void set_Attributes(
   System.IO.FileAttributes value
);
public System.DateTime get_CreationTime();
public void set_CreationTime(
   System.DateTime value
);
public abstract boolean get_Exists();
public System.String get_Extension();
public System.String get_FullName();
public System.DateTime get_LastAccessTime();
public void set_LastAccessTime(
   System.DateTime value
);
public System.DateTime get_LastWriteTime();
public void set_LastWriteTime(
   System.DateTime value
);
public abstract System.String get_Name();
```

Public Methods

```
public abstract void Delete();
public void Refresh();
```

Protected Constructors

```
protected FileSystemInfo();
```

Protected Fields

```
protected System.String FullPath;
protected System.String OriginalPath;
```

FileSystemWatcher Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Changed") */
public class FileSystemWatcher
   extends System.ComponentModel.Component
   implements System.ComponentModel.ISupportInitialize
```

Public Constructors

```
public FileSystemWatcher();
public FileSystemWatcher(
   System.String path
);
public FileSystemWatcher(
   System.String path,
   System.String filter
);
```

Public Properties

```
public boolean get_EnableRaisingEvents();
public void set_EnableRaisingEvents(
   boolean value
);
public System.String get_Filter();
public void set_Filter(
   System.String value
);
public boolean get_IncludeSubdirectories();
public void set_IncludeSubdirectories(
   boolean value
);
public int get_InternalBufferSize();
public void set_InternalBufferSize(
   int value
);
public System.IO.NotifyFilters get_NotifyFilter();
public void set_NotifyFilter(
   System.IO.NotifyFilters value
);
public System.String get_Path();
public void set_Path(
   System.String value
);
public System.ComponentModel.ISite get_Site();
public void set_Site(
   System.ComponentModel.ISite value
);
public System.ComponentModel.ISynchronizeInvoke get_SynchronizingObject();
public void set_SynchronizingObject(
   System.ComponentModel.ISynchronizeInvoke value
);
```

Public Methods

```
public final void BeginInit();
public final void EndInit();
public System.IO.WaitForChangedResult WaitForChanged(
   System.IO.WatcherChangeTypes changeType
);
public System.IO.WaitForChangedResult WaitForChanged(
   System.IO.WatcherChangeTypes changeType,
   int timeout
);
```

Public Events

```
public void add_Changed(
   System.IO.FileSystemEventHandler value
);
public void remove_Changed(
   System.IO.FileSystemEventHandler value
);
public void add_Created(
   System.IO.FileSystemEventHandler value
);
public void remove_Created(
   System.IO.FileSystemEventHandler value
);
public void add_Deleted(
   System.IO.FileSystemEventHandler value
);
public void remove_Deleted(
   System.IO.FileSystemEventHandler value
);
public void add_Error(
   System.IO.ErrorEventHandler value
);
public void remove_Error(
   System.IO.ErrorEventHandler value
);
public void add_Renamed(
   System.IO.RenamedEventHandler value
);
public void remove_Renamed(
   System.IO.RenamedEventHandler value
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void OnChanged(
   System.IO.FileSystemEventArgs e
);
protected void OnCreated(
   System.IO.FileSystemEventArgs e
);
protected void OnDeleted(
   System.IO.FileSystemEventArgs e
);
protected void OnError(
   System.IO.ErrorEventArgs e
);
protected void OnRenamed(
   System.IO.RenamedEventArgs e
);
```

InternalBufferOverflowException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class InternalBufferOverflowException
   extends System.SystemException
```

Public Constructors

```
public InternalBufferOverflowException();
public InternalBufferOverflowException(
   System.String message
);
public InternalBufferOverflowException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected InternalBufferOverflowException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

IODescriptionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class IODescriptionAttribute
   extends System.ComponentModel.DescriptionAttribute
```

Public Constructors

```
public IODescriptionAttribute(
   System.String description
);
```

Public Properties

```
public System.String get_Description();
```

IOException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class IOException
   extends System.SystemException
```

Public Constructors

```
public IOException();
public IOException(
   System.String message
);
public IOException(
   System.String message,
   int hresult
);
public IOException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected IOException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

MemoryStream Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class MemoryStream
   extends System.IO.Stream
```

Public Constructors

```
public MemoryStream();
public MemoryStream(
   ubyte[] buffer
);
public MemoryStream(
   int capacity
);
public MemoryStream(
   ubyte[] buffer,
   boolean writable
);
public MemoryStream(
   ubyte[] buffer,
   int index,
   int count
);
public MemoryStream(
   ubyte[] buffer,
   int index,
   int count,
   boolean writable
);
public MemoryStream(
   ubyte[] buffer,
   int index,
   int count,
   boolean writable,
   boolean publiclyVisible
);
```

Public Properties

```
public boolean get_CanRead();
public boolean get_CanSeek();
public boolean get_CanWrite();
public int get_Capacity();
public void set_Capacity(
   int value
);
public long get_Length();
public long get_Position();
public void set_Position(
   long value
);
```

Public Methods

```
public void Close();
public void Flush();
public ubyte[] GetBuffer();
public int Read(
   ubyte[] buffer,
   int offset,
   int count
);
public int ReadByte();
public long Seek(
   long offset,
   System.IO.SeekOrigin loc
);
public void SetLength(
   long value
);
public ubyte[] ToArray();
public void Write(
   ubyte[] buffer,
   int offset,
   int count
);
public void WriteByte(
   ubyte value
);
public void WriteTo(
   System.IO.Stream stream
);
```

NotifyFilters Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum NotifyFilters
```

Members

Attributes

CreationTime

DirectoryName

FileName

LastAccess

LastWrite

Security

Size

Path Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Path
   extends System.Object
```

Public Methods

```
public static System.String ChangeExtension(
   System.String path,
   System.String extension
);
public static System.String Combine(
   System.String path1,
   System.String path2
);
public static System.String GetDirectoryName(
   System.String path
);
public static System.String GetExtension(
   System.String path
);
public static System.String GetFileName(
   System.String path
);
public static System.String GetFileNameWithoutExtension(
   System.String path
);
public static System.String GetFullPath(
   System.String path
);
public static System.String GetPathRoot(
   System.String path
);
public static System.String GetTempFileName();
public static System.String GetTempPath();
public static boolean HasExtension(
   System.String path
);
public static boolean IsPathRooted(
   System.String path
);
```

Public Fields

```
public static final char AltDirectorySeparatorChar;
public static final char DirectorySeparatorChar;
public static final char[] InvalidPathChars;
public static final char PathSeparator;
public static final char VolumeSeparatorChar;
```

PathTooLongException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class PathTooLongException
   extends System.IO.IOException
```

Public Constructors

```
public PathTooLongException();
public PathTooLongException(
   System.String message
);
public PathTooLongException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected PathTooLongException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

RenamedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RenamedEventArgs
   extends System.IO.FileSystemEventArgs
```

Public Constructors

```
public RenamedEventArgs(
   System.IO.WatcherChangeTypes changeType,
   System.String directory,
   System.String name,
   System.String oldName
);
```

Public Properties

```
public System.String get_OldFullPath();
public System.String get_OldName();
```

RenamedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void RenamedEventHandler(
   System.Object sender,
   System.IO.RenamedEventArgs e
);
```

SeekOrigin Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SeekOrigin
```

Members

Begin

Current

End

Stream Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class Stream
   extends System.MarshalByRefObject
   implements System.IDisposable
```

Public Properties

```
public abstract boolean get_CanRead();
public abstract boolean get_CanSeek();
public abstract boolean get_CanWrite();
public abstract long get_Length();
public abstract long get_Position();
public abstract void set_Position(
   long value
);
```

Public Methods

```
public System.IAsyncResult BeginRead(
   ubyte[] buffer,
   int offset,
   int count,
   System.AsyncCallback callback,
   System.Object state
);
public System.IAsyncResult BeginWrite(
   ubyte[] buffer,
   int offset,
   int count,
   System.AsyncCallback callback,
   System.Object state
);
public void Close();
public int EndRead(
   System.IAsyncResult asyncResult
);
public void EndWrite(
   System.IAsyncResult asyncResult
);
public abstract void Flush();
public abstract int Read(
   ubyte[] buffer,
   int offset,
   int count
);
public int ReadByte();
public abstract long Seek(
   long offset,
   System.IO.SeekOrigin origin
);
public abstract void SetLength(
   long value
);
public abstract void Write(
   ubyte[] buffer,
```

```
    int offset,
    int count
);
public void WriteByte(
    ubyte value
);
```

Public Fields

```
public static final System.IO.Stream Null;
```

Protected Constructors

```
protected Stream();
```

Protected Methods

```
protected System.Threading.WaitHandle CreateWaitHandle();
```

StreamReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StreamReader
   extends System.IO.TextReader
```

Public Constructors

```
public StreamReader(
   System.IO.Stream stream
);
public StreamReader(
   System.String path
);
public StreamReader(
   System.IO.Stream stream,
   boolean detectEncodingFromByteOrderMarks
);
public StreamReader(
   System.IO.Stream stream,
   System.Text.Encoding encoding
);
public StreamReader(
   System.String path,
   boolean detectEncodingFromByteOrderMarks
);
public StreamReader(
   System.String path,
   System.Text.Encoding encoding
);
public StreamReader(
   System.IO.Stream stream,
   System.Text.Encoding encoding,
   boolean detectEncodingFromByteOrderMarks
);
public StreamReader(
   System.String path,
   System.Text.Encoding encoding,
   boolean detectEncodingFromByteOrderMarks
);
public StreamReader(
   System.IO.Stream stream,
   System.Text.Encoding encoding,
   boolean detectEncodingFromByteOrderMarks,
   int bufferSize
);
public StreamReader(
   System.String path,
   System.Text.Encoding encoding,
   boolean detectEncodingFromByteOrderMarks,
   int bufferSize
);
```

Public Properties

```
public System.IO.Stream get_BaseStream();
public System.Text.Encoding get_CurrentEncoding();
```

Public Methods

```
public void Close();
public void DiscardBufferedData();
public int Peek();
public int Read();
public int Read(
   char[] buffer,
   int index,
   int count
);
public System.String ReadLine();
public System.String ReadToEnd();
```

Public Fields

```
public static final System.IO.StreamReader Null;
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

StreamWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StreamWriter
   extends System.IO.TextWriter
```

Public Constructors

```
public StreamWriter(
   System.IO.Stream stream
);
public StreamWriter(
   System.String path
);
public StreamWriter(
   System.IO.Stream stream,
   System.Text.Encoding encoding
);
public StreamWriter(
   System.String path,
   boolean append
);
public StreamWriter(
   System.IO.Stream stream,
   System.Text.Encoding encoding,
   int bufferSize
);
public StreamWriter(
   System.String path,
   boolean append,
   System.Text.Encoding encoding
);
public StreamWriter(
   System.String path,
   boolean append,
   System.Text.Encoding encoding,
   int bufferSize
);
```

Public Properties

```
public boolean get_AutoFlush();
public void set_AutoFlush(
   boolean value
);
public System.IO.Stream get_BaseStream();
public System.Text.Encoding get_Encoding();
```

Public Methods

```
public void Close();
public void Flush();
public void Write(
   char value
);
```

```
public void Write(
   char[] buffer
);
public void Write(
   System.String value
);
public void Write(
   char[] buffer,
   int index,
   int count
);
```

Public Fields

```
public static final System.IO.StreamWriter Null;
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

StringReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StringReader
   extends System.IO.TextReader
```

Public Constructors

```
public StringReader(
   System.String s
);
```

Public Methods

```
public void Close();
public int Peek();
public int Read();
public int Read(
   char[] buffer,
   int index,
   int count
);
public System.String ReadLine();
public System.String ReadToEnd();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

StringWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StringWriter
   extends System.IO.TextWriter
```

Public Constructors

```
public StringWriter();
public StringWriter(
   System.IFormatProvider formatProvider
);
public StringWriter(
   System.Text.StringBuilder sb
);
public StringWriter(
   System.Text.StringBuilder sb,
   System.IFormatProvider formatProvider
);
```

Public Properties

```
public System.Text.Encoding get_Encoding();
```

Public Methods

```
public void Close();
public System.Text.StringBuilder GetStringBuilder();
public System.String ToString();
public void Write(
   char value
);
public void Write(
   System.String value
);
public void Write(
   char[] buffer,
   int index,
   int count
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

TextReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class TextReader
   extends System.MarshalByRefObject
   implements System.IDisposable
```

Public Methods

```
public void Close();
public int Peek();
public int Read();
public int Read(
   char[] buffer,
   int index,
   int count
);
public int ReadBlock(
   char[] buffer,
   int index,
   int count
);
public System.String ReadLine();
public System.String ReadToEnd();
public static System.IO.TextReader Synchronized(
   System.IO.TextReader reader
);
```

Public Fields

```
public static final System.IO.TextReader Null;
```

Protected Constructors

```
protected TextReader();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

TextWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class TextWriter
   extends System.MarshalByRefObject
   implements System.IDisposable
```

Public Properties

```
public abstract System.Text.Encoding get_Encoding();
public System.IFormatProvider get_FormatProvider();
public System.String get_NewLine();
public void set_NewLine(
   System.String value
);
```

Public Methods

```
public void Close();
public void Flush();
public static System.IO.TextWriter Synchronized(
   System.IO.TextWriter writer
);
public void Write(
   boolean value
);
public void Write(
   char value
);
public void Write(
   char[] buffer
);
public void Write(
   double value
);
public void Write(
   int value
);
public void Write(
   long value
);
public void Write(
   float value
);
public void Write(
   System.Decimal value
);
public void Write(
   System.Object value
);
public void Write(
   System.String value
);
public void Write(
   System.String format,
   System.Object arg0
);
```

```
public void Write(
   System.String format,
   System.Object[] arg
);
public void Write(
   char[] buffer,
   int index,
   int count
);
public void Write(
   System.String format,
   System.Object arg0,
   System.Object arg1
);
public void Write(
   System.String format,
   System.Object arg0,
   System.Object arg1,
   System.Object arg2
);
public void WriteLine();
public void WriteLine(
   boolean value
);
public void WriteLine(
   char value
);
public void WriteLine(
   char[] buffer
);
public void WriteLine(
   double value
);
public void WriteLine(
   int value
);
public void WriteLine(
   long value
);
public void WriteLine(
   float value
);
public void WriteLine(
   System.Decimal value
);
public void WriteLine(
   System.Object value
);
public void WriteLine(
   System.String value
);
public void WriteLine(
   System.String format,
   System.Object arg0
);
public void WriteLine(
   System.String format,
   System.Object[] arg
);
public void WriteLine(
   char[] buffer,
   int index,
   int count
);
public void WriteLine(
   System.String format,
   System.Object arg0,
   System.Object arg1
```

```
);
public void WriteLine(
   System.String format,
   System.Object arg0,
   System.Object arg1,
   System.Object arg2
);
```

Public Fields

```
public static final System.IO.TextWriter Null;
```

Protected Constructors

```
protected TextWriter();
protected TextWriter(
   System.IFormatProvider formatProvider
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

Protected Fields

```
protected char[] CoreNewLine;
```

WaitForChangedResult Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WaitForChangedResult
   extends System.ValueType
```

Public Properties

```
public System.IO.WatcherChangeTypes get_ChangeType();
public void set_ChangeType(
   System.IO.WatcherChangeTypes value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_OldName();
public void set_OldName(
   System.String value
);
public boolean get_TimedOut();
public void set_TimedOut(
   boolean value
);
```

WatcherChangeTypes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum WatcherChangeTypes
```

Members

All

Changed

Created

Deleted

Renamed

System.IO.IsolatedStorage Namespace (Visual J# Syntax)

Classes

IsolatedStorage

IsolatedStorageException

IsolatedStorageFile

IsolatedStorageFileStream

Interfaces

INormalizeForIsolatedStorage

Enumerations

IsolatedStorageScope

INormalizeForIsolatedStorage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface INormalizeForIsolatedStorage
```

Public Methods

```
System.Object Normalize();
```

IsolatedStorage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class IsolatedStorage
   extends System.MarshalByRefObject
```

Public Properties

```
public System.Object get_AssemblyIdentity();
public System.UInt64 get_CurrentSize();
public System.Object get_DomainIdentity();
public System.UInt64 get_MaximumSize();
public System.IO.IsolatedStorage.IsolatedStorageScope get_Scope();
```

Public Methods

```
public abstract void Remove();
```

Protected Constructors

```
protected IsolatedStorage();
```

Protected Properties

```
protected char get_SeparatorExternal();
protected char get_SeparatorInternal();
```

Protected Methods

```
protected abstract System.Security.Permissions.IsolatedStoragePermission GetPermission(
   System.Security.PermissionSet ps
);
protected void InitStore(
   System.IO.IsolatedStorage.IsolatedStorageScope scope,
   System.Type domainEvidenceType,
   System.Type assemblyEvidenceType
);
```

IsolatedStorageException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class IsolatedStorageException
   extends System.Exception
```

Public Constructors

```
public IsolatedStorageException();
public IsolatedStorageException(
   System.String message
);
public IsolatedStorageException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected IsolatedStorageException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

IsolatedStorageFile Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class IsolatedStorageFile
   extends System.IO.IsolatedStorage.IsolatedStorage
   implements System.IDisposable
```

Public Properties

```
public System.UInt64 get_CurrentSize();
public System.UInt64 get_MaximumSize();
```

Public Methods

```
public void Close();
public void CreateDirectory(
   System.String dir
);
public void DeleteDirectory(
   System.String dir
);
public void DeleteFile(
   System.String file
);
public final void Dispose();
public System.String[] GetDirectoryNames(
   System.String searchPattern
);
public static System.Collections.IEnumerator GetEnumerator(
   System.IO.IsolatedStorage.IsolatedStorageScope scope
);
public System.String[] GetFileNames(
   System.String searchPattern
);
public static System.IO.IsolatedStorage.IsolatedStorageFile GetStore(
   System.IO.IsolatedStorage.IsolatedStorageScope scope,
   System.Object domainIdentity,
   System.Object assemblyIdentity
);
public static System.IO.IsolatedStorage.IsolatedStorageFile GetStore(
   System.IO.IsolatedStorage.IsolatedStorageScope scope,
   System.Type domainEvidenceType,
   System.Type assemblyEvidenceType
);
public static System.IO.IsolatedStorage.IsolatedStorageFile GetStore(
   System.IO.IsolatedStorage.IsolatedStorageScope scope,
   System.Security.Policy.Evidence domainEvidence,
   System.Type domainEvidenceType,
   System.Security.Policy.Evidence assemblyEvidence,
   System.Type assemblyEvidenceType
);
public static System.IO.IsolatedStorage.IsolatedStorageFile GetUserStoreForAssembly();
public static System.IO.IsolatedStorage.IsolatedStorageFile GetUserStoreForDomain();
public void Remove();
public static void Remove(
   System.IO.IsolatedStorage.IsolatedStorageScope scope
);
```

Protected Methods

```
protected void Finalize();
protected System.Security.Permissions.IsolatedStoragePermission GetPermission(
   System.Security.PermissionSet ps
);
```

IsolatedStorageFileStream Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class IsolatedStorageFileStream
   extends System.IO.FileStream
```

Public Constructors

```
public IsolatedStorageFileStream(
   System.String path,
   System.IO.FileMode mode
);
public IsolatedStorageFileStream(
   System.String path,
   System.IO.FileMode mode,
   System.IO.FileAccess access
);
public IsolatedStorageFileStream(
   System.String path,
   System.IO.FileMode mode,
   System.IO.IsolatedStorage.IsolatedStorageFile isf
);
public IsolatedStorageFileStream(
   System.String path,
   System.IO.FileMode mode,
   System.IO.FileAccess access,
   System.IO.FileShare share
);
public IsolatedStorageFileStream(
   System.String path,
   System.IO.FileMode mode,
   System.IO.FileAccess access,
   System.IO.IsolatedStorage.IsolatedStorageFile isf
);
public IsolatedStorageFileStream(
   System.String path,
   System.IO.FileMode mode,
   System.IO.FileAccess access,
   System.IO.FileShare share,
   int bufferSize
);
public IsolatedStorageFileStream(
   System.String path,
   System.IO.FileMode mode,
   System.IO.FileAccess access,
   System.IO.FileShare share,
   System.IO.IsolatedStorage.IsolatedStorageFile isf
);
public IsolatedStorageFileStream(
   System.String path,
   System.IO.FileMode mode,
   System.IO.FileAccess access,
   System.IO.FileShare share,
   int bufferSize,
   System.IO.IsolatedStorage.IsolatedStorageFile isf
);
```

Public Properties

```
public boolean get_CanRead();
public boolean get_CanSeek();
public boolean get_CanWrite();
public System.IntPtr get_Handle();
public boolean get_IsAsync();
public long get_Length();
public long get_Position();
public void set_Position(
   long value
);
```

Public Methods

```
public System.IAsyncResult BeginRead(
   ubyte[] buffer,
   int offset,
   int numBytes,
   System.AsyncCallback userCallback,
   System.Object stateObject
);
public System.IAsyncResult BeginWrite(
   ubyte[] buffer,
   int offset,
   int numBytes,
   System.AsyncCallback userCallback,
   System.Object stateObject
);
public void Close();
public int EndRead(
   System.IAsyncResult asyncResult
);
public void EndWrite(
   System.IAsyncResult asyncResult
);
public void Flush();
public int Read(
   ubyte[] buffer,
   int offset,
   int count
);
public int ReadByte();
public long Seek(
   long offset,
   System.IO.SeekOrigin origin
);
public void SetLength(
   long value
);
public void Write(
   ubyte[] buffer,
   int offset,
   int count
);
public void WriteByte(
   ubyte value
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

IsolatedStorageScope Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum IsolatedStorageScope
```

Members

Assembly

Domain

None

Roaming

User

System.Management Namespace (Visual J# Syntax)

Classes

CompletedEventArgs

ConnectionOptions

DeleteOptions

EnumerationOptions

EventArrivedEventArgs

EventQuery

EventWatcherOptions

InvokeMethodOptions

ManagementBaseObject

ManagementClass

ManagementEventArgs

ManagementEventWatcher

ManagementException

ManagementNamedValueCollection

ManagementObject

ManagementObjectCollection

ManagementObjectCollection.ManagementObjectEnumerator

ManagementObjectSearcher

ManagementOperationObserver

ManagementOptions

ManagementPath

ManagementQuery

ManagementScope

MethodData

MethodDataCollection

MethodDataCollection.MethodDataEnumerator

ObjectGetOptions

ObjectPutEventArgs

ObjectQuery

ObjectReadyEventArgs

ProgressEventArgs

PropertyData

PropertyDataCollection

PropertyDataCollection.PropertyDataEnumerator

PutOptions

QualifierData

QualifierDataCollection

QualifierDataCollection.QualifierDataEnumerator

RelatedObjectQuery

RelationshipQuery

SelectQuery

StoppedEventArgs

WqlEventQuery

WqlObjectQuery

Delegates

CompletedEventHandler

EventArrivedEventHandler

ObjectPutEventHandler

ObjectReadyEventHandler

ProgressEventHandler

StoppedEventHandler

Enumerations

AuthenticationLevel

CimType

CodeLanguage

ComparisonSettings

ImpersonationLevel

ManagementStatus

PutType

TextFormat

AuthenticationLevel Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum AuthenticationLevel
```

Members

Call

Connect

Default

None

Packet

PacketIntegrity

PacketPrivacy

Unchanged

CimType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CimType
```

Members

Boolean

Char16

DateTime

Object

Real32

Real64

Reference

SInt16

SInt32

SInt64

SInt8

String

UInt16

UInt32

UInt64

UInt8

CodeLanguage Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CodeLanguage
```

Members

CSharp

JScript

VB

ComparisonSettings Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum ComparisonSettings
```

Members

IgnoreCase

IgnoreClass

IgnoreDefaultValues

IgnoreFlavor

IgnoreObjectSource

IgnoreQualifiers

IncludeAll

CompletedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CompletedEventArgs
   extends System.Management.ManagementEventArgs
```

Public Properties

```
public System.Management.ManagementStatus get_Status();
public System.Management.ManagementBaseObject get_StatusObject();
```

CompletedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void CompletedEventHandler(
   System.Object sender,
   System.Management.CompletedEventArgs e
);
```

ConnectionOptions Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ConnectionOptions
   extends System.Management.ManagementOptions
```

Public Constructors

```
public ConnectionOptions();
public ConnectionOptions(
   System.String locale,
   System.String username,
   System.String password,
   System.String authority,
   System.Management.ImpersonationLevel impersonation,
   System.Management.AuthenticationLevel authentication,
   boolean enablePrivileges,
   System.Management.ManagementNamedValueCollection context,
   System.TimeSpan timeout
);
```

Public Properties

```
public System.Management.AuthenticationLevel get_Authentication();
public void set_Authentication(
   System.Management.AuthenticationLevel value
);
public System.String get_Authority();
public void set_Authority(
   System.String value
);
public boolean get_EnablePrivileges();
public void set_EnablePrivileges(
   boolean value
);
public System.Management.ImpersonationLevel get_Impersonation();
public void set_Impersonation(
   System.Management.ImpersonationLevel value
);
public System.String get_Locale();
public void set_Locale(
   System.String value
);
public void set_Password(
   System.String value
);
public System.String get_Username();
public void set_Username(
   System.String value
);
```

Public Methods

```
public System.Object Clone();
```

DeleteOptions Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DeleteOptions
   extends System.Management.ManagementOptions
```

Public Constructors

```
public DeleteOptions();
public DeleteOptions(
   System.Management.ManagementNamedValueCollection context,
   System.TimeSpan timeout
);
```

Public Methods

```
public System.Object Clone();
```

EnumerationOptions Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EnumerationOptions
   extends System.Management.ManagementOptions
```

Public Constructors

```
public EnumerationOptions();
public EnumerationOptions(
   System.Management.ManagementNamedValueCollection context,
   System.TimeSpan timeout,
   int blockSize,
   boolean rewindable,
   boolean returnImmediatley,
   boolean useAmendedQualifiers,
   boolean ensureLocatable,
   boolean prototypeOnly,
   boolean directRead,
   boolean enumerateDeep
);
```

Public Properties

```
public int get_BlockSize();
public void set_BlockSize(
   int value
);
public boolean get_DirectRead();
public void set_DirectRead(
   boolean value
);
public boolean get_EnsureLocatable();
public void set_EnsureLocatable(
   boolean value
);
public boolean get_EnumerateDeep();
public void set_EnumerateDeep(
   boolean value
);
public boolean get_PrototypeOnly();
public void set_PrototypeOnly(
   boolean value
);
public boolean get_ReturnImmediately();
public void set_ReturnImmediately(
   boolean value
);
public boolean get_Rewindable();
public void set_Rewindable(
   boolean value
);
public boolean get_UseAmendedQualifiers();
public void set_UseAmendedQualifiers(
   boolean value
);
```

Public Methods

```
public System.Object Clone();
```

EventArrivedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EventArrivedEventArgs
   extends System.Management.ManagementEventArgs
```

Public Properties

```
public System.Management.ManagementBaseObject get_NewEvent();
```

EventArrivedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void EventArrivedEventHandler(
   System.Object sender,
   System.Management.EventArrivedEventArgs e
);
```

EventQuery Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EventQuery
   extends System.Management.ManagementQuery
```

Public Constructors

```
public EventQuery();
public EventQuery(
   System.String query
);
public EventQuery(
   System.String language,
   System.String query
);
```

Public Methods

```
public System.Object Clone();
```

EventWatcherOptions Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EventWatcherOptions
   extends System.Management.ManagementOptions
```

Public Constructors

```
public EventWatcherOptions();
public EventWatcherOptions(
   System.Management.ManagementNamedValueCollection context,
   System.TimeSpan timeout,
   int blockSize
);
```

Public Properties

```
public int get_BlockSize();
public void set_BlockSize(
   int value
);
```

Public Methods

```
public System.Object Clone();
```

ImpersonationLevel Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ImpersonationLevel
```

Members

Anonymous

Default

Delegate

Identify

Impersonate

InvokeMethodOptions Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InvokeMethodOptions
   extends System.Management.ManagementOptions
```

Public Constructors

```
public InvokeMethodOptions();
public InvokeMethodOptions(
   System.Management.ManagementNamedValueCollection context,
   System.TimeSpan timeout
);
```

Public Methods

```
public System.Object Clone();
```

ManagementBaseObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class ManagementBaseObject
   extends System.ComponentModel.Component
   implements System.ICloneable, System.Runtime.Serialization.ISerializable
```

Public Properties

```
public System.Management.ManagementPath get_ClassPath();
public System.Object get_Item(
   System.String propertyName
);
public void set_Item(
   System.String propertyName,
   System.Object value
);
public System.Management.PropertyDataCollection get_Properties();
public System.Management.QualifierDataCollection get_Qualifiers();
public System.Management.PropertyDataCollection get_SystemProperties();
```

Public Methods

```
public System.Object Clone();
public boolean CompareTo(
   System.Management.ManagementBaseObject otherObject,
   System.Management.ComparisonSettings settings
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public System.Object GetPropertyQualifierValue(
   System.String propertyName,
   System.String qualifierName
);
public System.Object GetPropertyValue(
   System.String propertyName
);
public System.Object GetQualifierValue(
   System.String qualifierName
);
public System.String GetText(
   System.Management.TextFormat format
);
public void SetPropertyQualifierValue(
   System.String propertyName,
   System.String qualifierName,
   System.Object qualifierValue
);
public void SetPropertyValue(
   System.String propertyName,
   System.Object propertyValue
);
public void SetQualifierValue(
```

```
   System.String qualifierName,
   System.Object qualifierValue
);
```

Public Operators

```
public static System.IntPtr op_Explicit(
   System.Management.ManagementBaseObject managementObject
);
```

Protected Constructors

```
protected ManagementBaseObject(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ManagementClass Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ManagementClass
   extends System.Management.ManagementObject
```

Public Constructors

```
public ManagementClass();
public ManagementClass(
   System.Management.ManagementPath path
);
public ManagementClass(
   System.String path
);
public ManagementClass(
   System.Management.ManagementPath path,
   System.Management.ObjectGetOptions options
);
public ManagementClass(
   System.String path,
   System.Management.ObjectGetOptions options
);
public ManagementClass(
   System.Management.ManagementScope scope,
   System.Management.ManagementPath path,
   System.Management.ObjectGetOptions options
);
public ManagementClass(
   System.String scope,
   System.String path,
   System.Management.ObjectGetOptions options
);
```

Public Properties

```
public System.Collections.Specialized.StringCollection get_Derivation();
public System.Management.MethodDataCollection get_Methods();
public System.Management.ManagementPath get_Path();
public void set_Path(
   System.Management.ManagementPath value
);
```

Public Methods

```
public System.Object Clone();
public System.Management.ManagementObject CreateInstance();
public System.Management.ManagementClass Derive(
   System.String newClassName
);
public System.Management.ManagementObjectCollection GetInstances();
public System.Management.ManagementObjectCollection GetInstances(
   System.Management.EnumerationOptions options
);
public void GetInstances(
```

```
   System.Management.ManagementOperationObserver watcher
);
public void GetInstances(
   System.Management.ManagementOperationObserver watcher,
   System.Management.EnumerationOptions options
);
public System.Management.ManagementObjectCollection GetRelatedClasses();
public System.Management.ManagementObjectCollection GetRelatedClasses(
   System.String relatedClass
);
public void GetRelatedClasses(
   System.Management.ManagementOperationObserver watcher
);
public void GetRelatedClasses(
   System.Management.ManagementOperationObserver watcher,
   System.String relatedClass
);
public System.Management.ManagementObjectCollection GetRelatedClasses(
   System.String relatedClass,
   System.String relationshipClass,
   System.String relationshipQualifier,
   System.String relatedQualifier,
   System.String relatedRole,
   System.String thisRole,
   System.Management.EnumerationOptions options
);
public void GetRelatedClasses(
   System.Management.ManagementOperationObserver watcher,
   System.String relatedClass,
   System.String relationshipClass,
   System.String relationshipQualifier,
   System.String relatedQualifier,
   System.String relatedRole,
   System.String thisRole,
   System.Management.EnumerationOptions options
);
public System.Management.ManagementObjectCollection GetRelationshipClasses();
public System.Management.ManagementObjectCollection GetRelationshipClasses(
   System.String relationshipClass
);
public void GetRelationshipClasses(
   System.Management.ManagementOperationObserver watcher
);
public void GetRelationshipClasses(
   System.Management.ManagementOperationObserver watcher,
   System.String relationshipClass
);
public System.Management.ManagementObjectCollection GetRelationshipClasses(
   System.String relationshipClass,
   System.String relationshipQualifier,
   System.String thisRole,
   System.Management.EnumerationOptions options
);
public void GetRelationshipClasses(
   System.Management.ManagementOperationObserver watcher,
   System.String relationshipClass,
   System.String relationshipQualifier,
   System.String thisRole,
   System.Management.EnumerationOptions options
);
public System.CodeDom.CodeTypeDeclaration GetStronglyTypedClassCode(
   boolean includeSystemClassInClassDef,
   boolean systemPropertyClass
);
public boolean GetStronglyTypedClassCode(
   System.Management.CodeLanguage lang,
   System.String filePath,
   System.String classNamespace
```

```
);
public System.Management.ManagementObjectCollection GetSubclasses();
public System.Management.ManagementObjectCollection GetSubclasses(
   System.Management.EnumerationOptions options
);
public void GetSubclasses(
   System.Management.ManagementOperationObserver watcher
);
public void GetSubclasses(
   System.Management.ManagementOperationObserver watcher,
   System.Management.EnumerationOptions options
);
```

ManagementEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class ManagementEventArgs
   extends System.EventArgs
```

Public Properties

```
public System.Object get_Context();
```

ManagementEventWatcher Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class ManagementEventWatcher
   extends System.ComponentModel.Component
```

Public Constructors

```
public ManagementEventWatcher();
public ManagementEventWatcher(
   System.Management.EventQuery query
);
public ManagementEventWatcher(
   System.String query
);
public ManagementEventWatcher(
   System.Management.ManagementScope scope,
   System.Management.EventQuery query
);
public ManagementEventWatcher(
   System.String scope,
   System.String query
);
public ManagementEventWatcher(
   System.Management.ManagementScope scope,
   System.Management.EventQuery query,
   System.Management.EventWatcherOptions options
);
public ManagementEventWatcher(
   System.String scope,
   System.String query,
   System.Management.EventWatcherOptions options
);
```

Public Properties

```
public System.Management.EventWatcherOptions get_Options();
public void set_Options(
   System.Management.EventWatcherOptions value
);
public System.Management.EventQuery get_Query();
public void set_Query(
   System.Management.EventQuery value
);
public System.Management.ManagementScope get_Scope();
public void set_Scope(
   System.Management.ManagementScope value
);
```

Public Methods

```
public void Start();
public void Stop();
public System.Management.ManagementBaseObject WaitForNextEvent();
```

Public Events

```
public void add_EventArrived(
   System.Management.EventArrivedEventHandler value
);
public void remove_EventArrived(
   System.Management.EventArrivedEventHandler value
);
public void add_Stopped(
   System.Management.StoppedEventHandler value
);
public void remove_Stopped(
   System.Management.StoppedEventHandler value
);
```

Protected Methods

```
protected void Finalize();
```

ManagementException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ManagementException
   extends System.SystemException
```

Public Properties

```
public System.Management.ManagementStatus get_ErrorCode();
public System.Management.ManagementBaseObject get_ErrorInformation();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected ManagementException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ManagementNamedValueCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ManagementNamedValueCollection
   extends System.Collections.Specialized.NameObjectCollectionBase
```

Public Constructors

```
public ManagementNamedValueCollection();
public ManagementNamedValueCollection(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Public Properties

```
public System.Object get_Item(
   System.String name
);
```

Public Methods

```
public void Add(
   System.String name,
   System.Object value
);
public System.Management.ManagementNamedValueCollection Clone();
public void Remove(
   System.String name
);
public void RemoveAll();
```

ManagementObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ManagementObject
   extends System.Management.ManagementBaseObject
```

Public Constructors

```
public ManagementObject();
public ManagementObject(
   System.Management.ManagementPath path
);
public ManagementObject(
   System.String path
);
public ManagementObject(
   System.Management.ManagementPath path,
   System.Management.ObjectGetOptions options
);
public ManagementObject(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public ManagementObject(
   System.String path,
   System.Management.ObjectGetOptions options
);
public ManagementObject(
   System.Management.ManagementScope scope,
   System.Management.ManagementPath path,
   System.Management.ObjectGetOptions options
);
public ManagementObject(
   System.String scopeString,
   System.String pathString,
   System.Management.ObjectGetOptions options
);
```

Public Properties

```
public System.Management.ManagementPath get_ClassPath();
public System.Management.ObjectGetOptions get_Options();
public void set_Options(
   System.Management.ObjectGetOptions value
);
public System.Management.ManagementPath get_Path();
public void set_Path(
   System.Management.ManagementPath value
);
public System.Management.ManagementScope get_Scope();
public void set_Scope(
   System.Management.ManagementScope value
);
```

Public Methods

```
public System.Object Clone();
public System.Management.ManagementPath CopyTo(
   System.Management.ManagementPath path
);
public System.Management.ManagementPath CopyTo(
   System.String path
);
public System.Management.ManagementPath CopyTo(
   System.Management.ManagementPath path,
   System.Management.PutOptions options
);
public System.Management.ManagementPath CopyTo(
   System.String path,
   System.Management.PutOptions options
);
public void CopyTo(
   System.Management.ManagementOperationObserver watcher,
   System.Management.ManagementPath path
);
public void CopyTo(
   System.Management.ManagementOperationObserver watcher,
   System.String path
);
public void CopyTo(
   System.Management.ManagementOperationObserver watcher,
   System.Management.ManagementPath path,
   System.Management.PutOptions options
);
public void CopyTo(
   System.Management.ManagementOperationObserver watcher,
   System.String path,
   System.Management.PutOptions options
);
public void Delete();
public void Delete(
   System.Management.DeleteOptions options
);
public void Delete(
   System.Management.ManagementOperationObserver watcher
);
public void Delete(
   System.Management.ManagementOperationObserver watcher,
   System.Management.DeleteOptions options
);
public void Get();
public void Get(
   System.Management.ManagementOperationObserver watcher
);
public System.Management.ManagementBaseObject GetMethodParameters(
   System.String methodName
);
public System.Management.ManagementObjectCollection GetRelated();
public System.Management.ManagementObjectCollection GetRelated(
   System.String relatedClass
);
public void GetRelated(
   System.Management.ManagementOperationObserver watcher
);
public void GetRelated(
   System.Management.ManagementOperationObserver watcher,
   System.String relatedClass
);
public System.Management.ManagementObjectCollection GetRelated(
   System.String relatedClass,
   System.String relationshipClass,
   System.String relationshipQualifier,
   System.String relatedQualifier,
   System.String relatedRole,
```

```
   System.String thisRole,
   boolean classDefinitionsOnly,
   System.Management.EnumerationOptions options
);
public void GetRelated(
   System.Management.ManagementOperationObserver watcher,
   System.String relatedClass,
   System.String relationshipClass,
   System.String relationshipQualifier,
   System.String relatedQualifier,
   System.String relatedRole,
   System.String thisRole,
   boolean classDefinitionsOnly,
   System.Management.EnumerationOptions options
);
public System.Management.ManagementObjectCollection GetRelationships();
public System.Management.ManagementObjectCollection GetRelationships(
   System.String relationshipClass
);
public void GetRelationships(
   System.Management.ManagementOperationObserver watcher
);
public void GetRelationships(
   System.Management.ManagementOperationObserver watcher,
   System.String relationshipClass
);
public System.Management.ManagementObjectCollection GetRelationships(
   System.String relationshipClass,
   System.String relationshipQualifier,
   System.String thisRole,
   boolean classDefinitionsOnly,
   System.Management.EnumerationOptions options
);
public void GetRelationships(
   System.Management.ManagementOperationObserver watcher,
   System.String relationshipClass,
   System.String relationshipQualifier,
   System.String thisRole,
   boolean classDefinitionsOnly,
   System.Management.EnumerationOptions options
);
public System.Object InvokeMethod(
   System.String methodName,
   System.Object[] args
);
public System.Management.ManagementBaseObject InvokeMethod(
   System.String methodName,
   System.Management.ManagementBaseObject inParameters,
   System.Management.InvokeMethodOptions options
);
public void InvokeMethod(
   System.Management.ManagementOperationObserver watcher,
   System.String methodName,
   System.Object[] args
);
public void InvokeMethod(
   System.Management.ManagementOperationObserver watcher,
   System.String methodName,
   System.Management.ManagementBaseObject inParameters,
   System.Management.InvokeMethodOptions options
);
public System.Management.ManagementPath Put();
public System.Management.ManagementPath Put(
   System.Management.PutOptions options
);
public void Put(
   System.Management.ManagementOperationObserver watcher
);
```

```
public void Put(
   System.Management.ManagementOperationObserver watcher,
   System.Management.PutOptions options
);
public System.String ToString();
```

ManagementObjectCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ManagementObjectCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable, System.IDisposa
ble
```

Public Properties

```
public final int get_Count();
public final boolean get_IsSynchronized();
public final System.Object get_SyncRoot();
```

Public Methods

```
public final void CopyTo(
   System.Array array,
   int index
);
public void CopyTo(
   System.Management.ManagementBaseObject[] objectCollection,
   int index
);
public final void Dispose();
public System.Management.ManagementObjectCollection+ManagementObjectEnumerator GetEnumerator(
);
```

Protected Methods

```
protected void Finalize();
```

ManagementObjectCollection.ManagementObjectEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ManagementObjectEnumerator
   extends System.Object
   implements System.Collections.IEnumerator, System.IDisposable
```

Public Properties

```
public System.Management.ManagementBaseObject get_Current();
```

Public Methods

```
public final void Dispose();
public final boolean MoveNext();
public final void Reset();
```

Protected Methods

```
protected void Finalize();
```

ManagementObjectSearcher Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class ManagementObjectSearcher
   extends System.ComponentModel.Component
```

Public Constructors

```
public ManagementObjectSearcher();
public ManagementObjectSearcher(
   System.Management.ObjectQuery query
);
public ManagementObjectSearcher(
   System.String queryString
);
public ManagementObjectSearcher(
   System.Management.ManagementScope scope,
   System.Management.ObjectQuery query
);
public ManagementObjectSearcher(
   System.String scope,
   System.String queryString
);
public ManagementObjectSearcher(
   System.Management.ManagementScope scope,
   System.Management.ObjectQuery query,
   System.Management.EnumerationOptions options
);
public ManagementObjectSearcher(
   System.String scope,
   System.String queryString,
   System.Management.EnumerationOptions options
);
```

Public Properties

```
public System.Management.EnumerationOptions get_Options();
public void set_Options(
   System.Management.EnumerationOptions value
);
public System.Management.ObjectQuery get_Query();
public void set_Query(
   System.Management.ObjectQuery value
);
public System.Management.ManagementScope get_Scope();
public void set_Scope(
   System.Management.ManagementScope value
);
```

Public Methods

```
public System.Management.ManagementObjectCollection Get();
public void Get(
   System.Management.ManagementOperationObserver watcher
);
```

ManagementOperationObserver Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ManagementOperationObserver
   extends System.Object
```

Public Constructors

```
public ManagementOperationObserver();
```

Public Methods

```
public void Cancel();
```

Public Events

```
public void add_Completed(
   System.Management.CompletedEventHandler value
);
public void remove_Completed(
   System.Management.CompletedEventHandler value
);
public void add_ObjectPut(
   System.Management.ObjectPutEventHandler value
);
public void remove_ObjectPut(
   System.Management.ObjectPutEventHandler value
);
public void add_ObjectReady(
   System.Management.ObjectReadyEventHandler value
);
public void remove_ObjectReady(
   System.Management.ObjectReadyEventHandler value
);
public void add_Progress(
   System.Management.ProgressEventHandler value
);
public void remove_Progress(
   System.Management.ProgressEventHandler value
);
```

ManagementOptions Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Compo
nentModel.ExpandableObjectConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyTo
ken=b77a5c561934e089") */
public abstract class ManagementOptions
   extends System.Object
   implements System.ICloneable
```

Public Properties

```
public System.Management.ManagementNamedValueCollection get_Context();
public void set_Context(
   System.Management.ManagementNamedValueCollection value
);
public System.TimeSpan get_Timeout();
public void set_Timeout(
   System.TimeSpan value
);
```

Public Methods

```
public abstract System.Object Clone();
```

Public Fields

```
public static final System.TimeSpan InfiniteTimeout;
```

ManagementPath Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Manag
ement.ManagementPathConverter, System.Management, Version=1.0.3300.0, Culture=neutral, Public
KeyToken=b03f5f7f11d50a3a") */
public class ManagementPath
   extends System.Object
   implements System.ICloneable
```

Public Constructors

```
public ManagementPath();
public ManagementPath(
   System.String path
);
```

Public Properties

```
public System.String get_ClassName();
public void set_ClassName(
   System.String value
);
public static System.Management.ManagementPath get_DefaultPath();
public static void set_DefaultPath(
   System.Management.ManagementPath value
);
public boolean get_IsClass();
public boolean get_IsInstance();
public boolean get_IsSingleton();
public System.String get_NamespacePath();
public void set_NamespacePath(
   System.String value
);
public System.String get_Path();
public void set_Path(
   System.String value
);
public System.String get_RelativePath();
public void set_RelativePath(
   System.String value
);
public System.String get_Server();
public void set_Server(
   System.String value
);
```

Public Methods

```
public System.Management.ManagementPath Clone();
public void SetAsClass();
public void SetAsSingleton();
public System.String ToString();
```

ManagementQuery Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Manag
ement.ManagementQueryConverter, System.Management, Version=1.0.3300.0, Culture=neutral, Publi
cKeyToken=b03f5f7f11d50a3a") */
public abstract class ManagementQuery
   extends System.Object
   implements System.ICloneable
```

Public Properties

```
public System.String get_QueryLanguage();
public void set_QueryLanguage(
   System.String value
);
public System.String get_QueryString();
public void set_QueryString(
   System.String value
);
```

Public Methods

```
public abstract System.Object Clone();
```

Protected Methods

```
protected System.String GetValueOfKeyword(
   System.String query,
   System.String keyword,
   int keywordIndex
);
protected void ParseQuery(
   System.String query
);
```

ManagementScope Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Compo
nentModel.ExpandableObjectConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyTo
ken=b77a5c561934e089") */
public class ManagementScope
   extends System.Object
   implements System.ICloneable
```

Public Constructors

```
public ManagementScope();
public ManagementScope(
   System.Management.ManagementPath path
);
public ManagementScope(
   System.String path
);
public ManagementScope(
   System.Management.ManagementPath path,
   System.Management.ConnectionOptions options
);
public ManagementScope(
   System.String path,
   System.Management.ConnectionOptions options
);
```

Public Properties

```
public boolean get_IsConnected();
public System.Management.ConnectionOptions get_Options();
public void set_Options(
   System.Management.ConnectionOptions value
);
public System.Management.ManagementPath get_Path();
public void set_Path(
   System.Management.ManagementPath value
);
```

Public Methods

```
public System.Management.ManagementScope Clone();
public void Connect();
```

ManagementStatus Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ManagementStatus
```

Members

AccessDenied

AggregatingByObject

AlreadyExists

AmendedObject

BackupRestoreWinmgmtRunning

BufferTooSmall

CallCanceled

CannotBeAbstract

CannotBeKey

CannotBeSingleton

CannotChangeIndexInheritance

CannotChangeKeyInheritance

CircularReference

ClassHasChildren

ClassHasInstances

ClientTooSlow

CriticalError

Different

DuplicateObjects

Failed

False

IllegalNull

IllegalOperation

IncompleteClass

InitializationFailure

InvalidCimType

InvalidClass

InvalidContext

InvalidDuplicateParameter

InvalidFlavor

InvalidMethod

InvalidMethodParameters

InvalidNamespace

InvalidObject

InvalidObjectPath

InvalidOperation

InvalidOperator

InvalidParameter

InvalidParameterID

InvalidProperty

InvalidPropertyType

InvalidProviderRegistration

InvalidQualifier

InvalidQualifierType

InvalidQuery

InvalidQueryType

InvalidStream

InvalidSuperclass

InvalidSyntax

LocalCredentials

MarshalInvalidSignature

MarshalVersionMismatch

MethodDisabled

MethodNotImplemented

MissingAggregationList

MissingGroupWithin

MissingParameterID

NoError

NoMoreData

NonconsecutiveParameterIDs

NondecoratedObject

NotAvailable

NotEventClass

NotFound

NotSupported

OperationCanceled

OutOfDiskSpace

OutOfMemory

OverrideNotAllowed

ParameterIDOnRetval

PartialResults

Pending

PrivilegeNotHeld

PropagatedMethod

PropagatedProperty

PropagatedQualifier

PropertyNotAnObject

ProviderFailure

ProviderLoadFailure

ProviderNotCapable

ProviderNotFound

QueryNotImplemented

QueueOverflow

ReadOnly

RefresherBusy

RegistrationTooBroad

RegistrationTooPrecise

ResetToDefault

ServerTooBusy

ShuttingDown

SystemProperty

Timedout

TooManyProperties

TooMuchData

TransportFailure

TypeMismatch

Unexpected

UninterpretableProviderQuery

UnknownObjectType

UnknownPacketType

UnparsableQuery

UnsupportedClassUpdate

UnsupportedParameter

UnsupportedPutExtension

UpdateOverrideNotAllowed

UpdatePropagatedMethod

UpdateTypeMismatch

ValueOutOfRange

MethodData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MethodData
   extends System.Object
```

Public Properties

```
public System.Management.ManagementBaseObject get_InParameters();
public System.String get_Name();
public System.String get_Origin();
public System.Management.ManagementBaseObject get_OutParameters();
public System.Management.QualifierDataCollection get_Qualifiers();
```

MethodDataCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class MethodDataCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public final boolean get_IsSynchronized();
public System.Management.MethodData get_Item(
   System.String methodName
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public void Add(
   System.String methodName
);
public void Add(
   System.String methodName,
   System.Management.ManagementBaseObject inParams,
   System.Management.ManagementBaseObject outParams
);
public final void CopyTo(
   System.Array array,
   int index
);
public void CopyTo(
   System.Management.MethodData[] methodArray,
   int index
);
public System.Management.MethodDataCollection+MethodDataEnumerator GetEnumerator();
public void Remove(
   System.String methodName
);
```

MethodDataCollection.MethodDataEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MethodDataEnumerator
   extends System.Object
   implements System.Collections.IEnumerator
```

Public Properties

```
public System.Management.MethodData get_Current();
```

Public Methods

```
public final boolean MoveNext();
public final void Reset();
```

ObjectGetOptions Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ObjectGetOptions
   extends System.Management.ManagementOptions
```

Public Constructors

```
public ObjectGetOptions();
public ObjectGetOptions(
   System.Management.ManagementNamedValueCollection context
);
public ObjectGetOptions(
   System.Management.ManagementNamedValueCollection context,
   System.TimeSpan timeout,
   boolean useAmendedQualifiers
);
```

Public Properties

```
public boolean get_UseAmendedQualifiers();
public void set_UseAmendedQualifiers(
   boolean value
);
```

Public Methods

```
public System.Object Clone();
```

ObjectPutEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ObjectPutEventArgs
   extends System.Management.ManagementEventArgs
```

Public Properties

```
public System.Management.ManagementPath get_Path();
```

ObjectPutEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ObjectPutEventHandler(
   System.Object sender,
   System.Management.ObjectPutEventArgs e
);
```

ObjectQuery Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ObjectQuery
   extends System.Management.ManagementQuery
```

Public Constructors

```
public ObjectQuery();
public ObjectQuery(
   System.String query
);
public ObjectQuery(
   System.String language,
   System.String query
);
```

Public Methods

```
public System.Object Clone();
```

ObjectReadyEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ObjectReadyEventArgs
   extends System.Management.ManagementEventArgs
```

Public Properties

```
public System.Management.ManagementBaseObject get_NewObject();
```

ObjectReadyEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ObjectReadyEventHandler(
   System.Object sender,
   System.Management.ObjectReadyEventArgs e
);
```

ProgressEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ProgressEventArgs
   extends System.Management.ManagementEventArgs
```

Public Properties

```
public int get_Current();
public System.String get_Message();
public int get_UpperBound();
```

ProgressEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ProgressEventHandler(
   System.Object sender,
   System.Management.ProgressEventArgs e
);
```

PropertyData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PropertyData
   extends System.Object
```

Public Properties

```
public boolean get_IsArray();
public boolean get_IsLocal();
public System.String get_Name();
public System.String get_Origin();
public System.Management.QualifierDataCollection get_Qualifiers();
public System.Management.CimType get_Type();
public System.Object get_Value();
public void set_Value(
   System.Object value
);
```

PropertyDataCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class PropertyDataCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public final boolean get_IsSynchronized();
public System.Management.PropertyData get_Item(
   System.String propertyName
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public void Add(
   System.String propertyName,
   System.Object propertyValue
);
public void Add(
   System.String propertyName,
   System.Management.CimType propertyType,
   boolean isArray
);
public void Add(
   System.String propertyName,
   System.Object propertyValue,
   System.Management.CimType propertyType
);
public final void CopyTo(
   System.Array array,
   int index
);
public void CopyTo(
   System.Management.PropertyData[] propertyArray,
   int index
);
public System.Management.PropertyDataCollection+PropertyDataEnumerator GetEnumerator();
public void Remove(
   System.String propertyName
);
```

PropertyDataCollection.PropertyDataEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PropertyDataEnumerator
   extends System.Object
   implements System.Collections.IEnumerator
```

Public Properties

```
public System.Management.PropertyData get_Current();
```

Public Methods

```
public final boolean MoveNext();
public final void Reset();
```

PutOptions Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PutOptions
   extends System.Management.ManagementOptions
```

Public Constructors

```
public PutOptions();
public PutOptions(
   System.Management.ManagementNamedValueCollection context
);
public PutOptions(
   System.Management.ManagementNamedValueCollection context,
   System.TimeSpan timeout,
   boolean useAmendedQualifiers,
   System.Management.PutType putType
);
```

Public Properties

```
public System.Management.PutType get_Type();
public void set_Type(
   System.Management.PutType value
);
public boolean get_UseAmendedQualifiers();
public void set_UseAmendedQualifiers(
   boolean value
);
```

Public Methods

```
public System.Object Clone();
```

PutType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PutType
```

Members

CreateOnly

UpdateOnly

UpdateOrCreate

QualifierData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class QualifierData
   extends System.Object
```

Public Properties

```
public boolean get_IsAmended();
public void set_IsAmended(
   boolean value
);
public boolean get_IsLocal();
public boolean get_IsOverridable();
public void set_IsOverridable(
   boolean value
);
public System.String get_Name();
public boolean get_PropagatesToInstance();
public void set_PropagatesToInstance(
   boolean value
);
public boolean get_PropagatesToSubclass();
public void set_PropagatesToSubclass(
   boolean value
);
public System.Object get_Value();
public void set_Value(
   System.Object value
);
```

QualifierDataCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class QualifierDataCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public final boolean get_IsSynchronized();
public System.Management.QualifierData get_Item(
   System.String qualifierName
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public void Add(
   System.String qualifierName,
   System.Object qualifierValue
);
public void Add(
   System.String qualifierName,
   System.Object qualifierValue,
   boolean isAmended,
   boolean propagatesToInstance,
   boolean propagatesToSubclass,
   boolean isOverridable
);
public final void CopyTo(
   System.Array array,
   int index
);
public void CopyTo(
   System.Management.QualifierData[] qualifierArray,
   int index
);
public System.Management.QualifierDataCollection+QualifierDataEnumerator GetEnumerator();
public void Remove(
   System.String qualifierName
);
```

QualifierDataCollection.QualifierDataEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class QualifierDataEnumerator
   extends System.Object
   implements System.Collections.IEnumerator
```

Public Properties

```
public System.Management.QualifierData get_Current();
```

Public Methods

```
public final boolean MoveNext();
public final void Reset();
```

RelatedObjectQuery Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RelatedObjectQuery
   extends System.Management.WqlObjectQuery
```

Public Constructors

```
public RelatedObjectQuery();
public RelatedObjectQuery(
   System.String queryOrSourceObject
);
public RelatedObjectQuery(
   System.String sourceObject,
   System.String relatedClass
);
public RelatedObjectQuery(
   boolean isSchemaQuery,
   System.String sourceObject,
   System.String relatedClass,
   System.String relationshipClass,
   System.String relatedQualifier,
   System.String relationshipQualifier,
   System.String relatedRole,
   System.String thisRole
);
public RelatedObjectQuery(
   System.String sourceObject,
   System.String relatedClass,
   System.String relationshipClass,
   System.String relatedQualifier,
   System.String relationshipQualifier,
   System.String relatedRole,
   System.String thisRole,
   boolean classDefinitionsOnly
);
```

Public Properties

```
public boolean get_ClassDefinitionsOnly();
public void set_ClassDefinitionsOnly(
   boolean value
);
public boolean get_IsSchemaQuery();
public void set_IsSchemaQuery(
   boolean value
);
public System.String get_RelatedClass();
public void set_RelatedClass(
   System.String value
);
public System.String get_RelatedQualifier();
public void set_RelatedQualifier(
   System.String value
);
public System.String get_RelatedRole();
public void set_RelatedRole(
   System.String value
```

```
);
public System.String get_RelationshipClass();
public void set_RelationshipClass(
   System.String value
);
public System.String get_RelationshipQualifier();
public void set_RelationshipQualifier(
   System.String value
);
public System.String get_SourceObject();
public void set_SourceObject(
   System.String value
);
public System.String get_ThisRole();
public void set_ThisRole(
   System.String value
);
```

Public Methods

```
public System.Object Clone();
```

Protected Methods

```
protected void BuildQuery();
protected void ParseQuery(
   System.String query
);
```

RelationshipQuery Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RelationshipQuery
   extends System.Management.WqlObjectQuery
```

Public Constructors

```
public RelationshipQuery();
public RelationshipQuery(
   System.String queryOrSourceObject
);
public RelationshipQuery(
   System.String sourceObject,
   System.String relationshipClass
);
public RelationshipQuery(
   boolean isSchemaQuery,
   System.String sourceObject,
   System.String relationshipClass,
   System.String relationshipQualifier,
   System.String thisRole
);
public RelationshipQuery(
   System.String sourceObject,
   System.String relationshipClass,
   System.String relationshipQualifier,
   System.String thisRole,
   boolean classDefinitionsOnly
);
```

Public Properties

```
public boolean get_ClassDefinitionsOnly();
public void set_ClassDefinitionsOnly(
   boolean value
);
public boolean get_IsSchemaQuery();
public void set_IsSchemaQuery(
   boolean value
);
public System.String get_RelationshipClass();
public void set_RelationshipClass(
   System.String value
);
public System.String get_RelationshipQualifier();
public void set_RelationshipQualifier(
   System.String value
);
public System.String get_SourceObject();
public void set_SourceObject(
   System.String value
);
public System.String get_ThisRole();
public void set_ThisRole(
   System.String value
);
```

Public Methods

```
public System.Object Clone();
```

Protected Methods

```
protected void BuildQuery();
protected void ParseQuery(
   System.String query
);
```

SelectQuery Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SelectQuery
   extends System.Management.WqlObjectQuery
```

Public Constructors

```
public SelectQuery();
public SelectQuery(
   System.String queryOrClassName
);
public SelectQuery(
   boolean isSchemaQuery,
   System.String condition
);
public SelectQuery(
   System.String className,
   System.String condition
);
public SelectQuery(
   System.String className,
   System.String condition,
   System.String[] selectedProperties
);
```

Public Properties

```
public System.String get_ClassName();
public void set_ClassName(
   System.String value
);
public System.String get_Condition();
public void set_Condition(
   System.String value
);
public boolean get_IsSchemaQuery();
public void set_IsSchemaQuery(
   boolean value
);
public System.String get_QueryString();
public void set_QueryString(
   System.String value
);
public System.Collections.Specialized.StringCollection get_SelectedProperties();
public void set_SelectedProperties(
   System.Collections.Specialized.StringCollection value
);
```

Public Methods

```
public System.Object Clone();
```

Protected Methods

```
protected void BuildQuery();
protected void ParseQuery(
   System.String query
);
```

StoppedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class StoppedEventArgs
   extends System.Management.ManagementEventArgs
```

Public Properties

```
public System.Management.ManagementStatus get_Status();
```

StoppedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void StoppedEventHandler(
   System.Object sender,
   System.Management.StoppedEventArgs e
);
```

TextFormat Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TextFormat
```

Members

Mof

WqlEventQuery Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WqlEventQuery
   extends System.Management.EventQuery
```

Public Constructors

```
public WqlEventQuery();
public WqlEventQuery(
   System.String queryOrEventClassName
);
public WqlEventQuery(
   System.String eventClassName,
   System.String condition
);
public WqlEventQuery(
   System.String eventClassName,
   System.TimeSpan withinInterval
);
public WqlEventQuery(
   System.String eventClassName,
   System.String condition,
   System.TimeSpan groupWithinInterval
);
public WqlEventQuery(
   System.String eventClassName,
   System.TimeSpan withinInterval,
   System.String condition
);
public WqlEventQuery(
   System.String eventClassName,
   System.String condition,
   System.TimeSpan groupWithinInterval,
   System.String[] groupByPropertyList
);
public WqlEventQuery(
   System.String eventClassName,
   System.TimeSpan withinInterval,
   System.String condition,
   System.TimeSpan groupWithinInterval,
   System.String[] groupByPropertyList,
   System.String havingCondition
);
```

Public Properties

```
public System.String get_Condition();
public void set_Condition(
   System.String value
);
public System.String get_EventClassName();
public void set_EventClassName(
   System.String value
);
public System.Collections.Specialized.StringCollection get_GroupByPropertyList();
public void set_GroupByPropertyList(
   System.Collections.Specialized.StringCollection value
```

```
);
public System.TimeSpan get_GroupWithinInterval();
public void set_GroupWithinInterval(
   System.TimeSpan value
);
public System.String get_HavingCondition();
public void set_HavingCondition(
   System.String value
);
public System.String get_QueryLanguage();
public System.String get_QueryString();
public void set_QueryString(
   System.String value
);
public System.TimeSpan get_WithinInterval();
public void set_WithinInterval(
   System.TimeSpan value
);
```

Public Methods

```
public System.Object Clone();
```

Protected Methods

```
protected void BuildQuery();
protected void ParseQuery(
   System.String query
);
```

WqlObjectQuery Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WqlObjectQuery
   extends System.Management.ObjectQuery
```

Public Constructors

```
public WqlObjectQuery();
public WqlObjectQuery(
   System.String query
);
```

Public Properties

```
public System.String get_QueryLanguage();
```

Public Methods

```
public System.Object Clone();
```

System.Management.Instrumentation Namespace (Visual J# Syntax)

Classes

BaseEvent

DefaultManagementProjectInstaller

IgnoreMemberAttribute

Instance

Instrumentation

InstrumentationClassAttribute

InstrumentedAttribute

ManagedNameAttribute

ManagementInstaller

Interfaces

IEvent

IInstance

Enumerations

InstrumentationType

BaseEvent Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Management.Instrumentation.InstrumentationClassAttribute(Instrumentatio
nType = Event) */
public abstract class BaseEvent
   extends System.Object
   implements System.Management.Instrumentation.IEvent
```

Public Methods

```
public final void Fire();
```

Protected Constructors

```
protected BaseEvent();
```

DefaultManagementProjectInstaller Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DefaultManagementProjectInstaller
   extends System.Configuration.Install.Installer
```

Public Constructors

```
public DefaultManagementProjectInstaller();
```

IEvent Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IEvent
```

Public Methods

```
void Fire();
```

IgnoreMemberAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, Property, Field, AllowMultipl
e = false, Inherited = true) */
public class IgnoreMemberAttribute
   extends System.Attribute
```

Public Constructors

```
public IgnoreMemberAttribute();
```

IInstance Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IInstance
```

Public Properties

```
boolean get_Published();
void set_Published(
   boolean value
);
```

Instance Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Management.Instrumentation.InstrumentationClassAttribute(Instrumentatio
nType = Instance) */
public abstract class Instance
   extends System.Object
   implements System.Management.Instrumentation.IInstance
```

Public Properties

```
public final boolean get_Published();
public final void set_Published(
   boolean value
);
```

Protected Constructors

```
protected Instance();
```

Instrumentation Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Instrumentation
   extends System.Object
```

Public Constructors

```
public Instrumentation();
```

Public Methods

```
public static void Fire(
   System.Object eventData
);
public static boolean IsAssemblyRegistered(
   System.Reflection.Assembly assemblyToRegister
);
public static void Publish(
   System.Object instanceData
);
public static void RegisterAssembly(
   System.Reflection.Assembly assemblyToRegister
);
public static void Revoke(
   System.Object instanceData
);
public static void SetBatchSize(
   System.Type instrumentationClass,
   int batchSize
);
```

InstrumentationClassAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, AllowMultiple = false,
Inherited = true) */
public class InstrumentationClassAttribute
   extends System.Attribute
```

Public Constructors

```
public InstrumentationClassAttribute(
   System.Management.Instrumentation.InstrumentationType instrumentationType
);
public InstrumentationClassAttribute(
   System.Management.Instrumentation.InstrumentationType instrumentationType,
   System.String managedBaseClassName
);
```

Public Properties

```
public System.Management.Instrumentation.InstrumentationType get_InstrumentationType();
public System.String get_ManagedBaseClassName();
```

InstrumentationType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum InstrumentationType
```

Members

Abstract

Event

Instance

InstrumentedAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class InstrumentedAttribute
   extends System.Attribute
```

Public Constructors

```
public InstrumentedAttribute();
public InstrumentedAttribute(
   System.String namespaceName
);
public InstrumentedAttribute(
   System.String namespaceName,
   System.String securityDescriptor
);
```

Public Properties

```
public System.String get_NamespaceName();
public System.String get_SecurityDescriptor();
```

ManagedNameAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, Method, Property, Fiel
d, AllowMultiple = false, Inherited = true) */
public class ManagedNameAttribute
   extends System.Attribute
```

Public Constructors

```
public ManagedNameAttribute(
   System.String name
);
```

ManagementInstaller Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ManagementInstaller
   extends System.Configuration.Install.Installer
```

Public Constructors

```
public ManagementInstaller();
```

Public Properties

```
public System.String get_HelpText();
```

Public Methods

```
public void Commit(
   System.Collections.IDictionary savedState
);
public void Install(
   System.Collections.IDictionary savedState
);
public void Rollback(
   System.Collections.IDictionary savedState
);
public void Uninstall(
   System.Collections.IDictionary savedState
);
```

System.Messaging Namespace (Visual J# Syntax)

Classes

AccessControlEntry

AccessControlList

ActiveXMessageFormatter

BinaryMessageFormatter

DefaultPropertiesToSend

Message

MessageEnumerator

MessagePropertyFilter

MessageQueue

MessageQueueAccessControlEntry

MessageQueueCriteria

MessageQueueEnumerator

MessageQueueException

MessageQueueInstaller

MessageQueuePermission

MessageQueuePermissionAttribute

MessageQueuePermissionEntry

MessageQueuePermissionEntryCollection

MessageQueueTransaction

MessagingDescriptionAttribute

PeekCompletedEventArgs

ReceiveCompletedEventArgs

Trustee

XmlMessageFormatter

Interfaces

IMessageFormatter

Delegates

PeekCompletedEventHandler

ReceiveCompletedEventHandler

Enumerations

AccessControlEntryType

AcknowledgeTypes

Acknowledgment

CryptographicProviderType

EncryptionAlgorithm

EncryptionRequired

GenericAccessRights

HashAlgorithm

MessagePriority

MessageQueueAccessRights

MessageQueueErrorCode

MessageQueuePermissionAccess

MessageQueueTransactionStatus

MessageQueueTransactionType

MessageType

StandardAccessRights

TrusteeType

AccessControlEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AccessControlEntry
   extends System.Object
```

Public Constructors

```
public AccessControlEntry();
public AccessControlEntry(
   System.Messaging.Trustee trustee
);
public AccessControlEntry(
   System.Messaging.Trustee trustee,
   System.Messaging.GenericAccessRights genericAccessRights,
   System.Messaging.StandardAccessRights standardAccessRights,
   System.Messaging.AccessControlEntryType entryType
);
```

Public Properties

```
public System.Messaging.AccessControlEntryType get_EntryType();
public void set_EntryType(
   System.Messaging.AccessControlEntryType value
);
public System.Messaging.GenericAccessRights get_GenericAccessRights();
public void set_GenericAccessRights(
   System.Messaging.GenericAccessRights value
);
public System.Messaging.StandardAccessRights get_StandardAccessRights();
public void set_StandardAccessRights(
   System.Messaging.StandardAccessRights value
);
public System.Messaging.Trustee get_Trustee();
public void set_Trustee(
   System.Messaging.Trustee value
);
```

Protected Properties

```
protected int get_CustomAccessRights();
protected void set_CustomAccessRights(
   int value
);
```

AccessControlEntryType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum AccessControlEntryType
```

Members

Allow

Deny

Revoke

Set

AccessControlList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AccessControlList
   extends System.Collections.CollectionBase
```

Public Constructors

```
public AccessControlList();
```

Public Methods

```
public int Add(
   System.Messaging.AccessControlEntry entry
);
public boolean Contains(
   System.Messaging.AccessControlEntry entry
);
public void CopyTo(
   System.Messaging.AccessControlEntry[] array,
   int index
);
public int IndexOf(
   System.Messaging.AccessControlEntry entry
);
public void Insert(
   int index,
   System.Messaging.AccessControlEntry entry
);
public void Remove(
   System.Messaging.AccessControlEntry entry
);
```

AcknowledgeTypes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum AcknowledgeTypes
```

Members

FullReachQueue

FullReceive

NegativeReceive

None

NotAcknowledgeReachQueue

NotAcknowledgeReceive

PositiveArrival

PositiveReceive

Acknowledgment Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum Acknowledgment
```

Members

AccessDenied

BadDestinationQueue

BadEncryption

BadSignature

CouldNotEncrypt

HopCountExceeded

None

NotTransactionalMessage

NotTransactionalQueue

Purged

QueueDeleted

QueueExceedMaximumSize

QueuePurged

ReachQueue

ReachQueueTimeout

Receive

ReceiveTimeout

ActiveXMessageFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ActiveXMessageFormatter
   extends System.Object
   implements System.Messaging.IMessageFormatter, System.ICloneable
```

Public Constructors

```
public ActiveXMessageFormatter();
```

Public Methods

```
public final boolean CanRead(
   System.Messaging.Message message
);
public final System.Object Clone();
public static void InitStreamedObject(
   System.Object streamedObject
);
public final System.Object Read(
   System.Messaging.Message message
);
public final void Write(
   System.Messaging.Message message,
   System.Object obj
);
```

BinaryMessageFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BinaryMessageFormatter
   extends System.Object
   implements System.Messaging.IMessageFormatter, System.ICloneable
```

Public Constructors

```
public BinaryMessageFormatter();
public BinaryMessageFormatter(
   System.Runtime.Serialization.Formatters.FormatterAssemblyStyle topObjectFormat,
   System.Runtime.Serialization.Formatters.FormatterTypeStyle typeFormat
);
```

Public Properties

```
public System.Runtime.Serialization.Formatters.FormatterAssemblyStyle get_TopObjectFormat();
public void set_TopObjectFormat(
   System.Runtime.Serialization.Formatters.FormatterAssemblyStyle value
);
public System.Runtime.Serialization.Formatters.FormatterTypeStyle get_TypeFormat();
public void set_TypeFormat(
   System.Runtime.Serialization.Formatters.FormatterTypeStyle value
);
```

Public Methods

```
public final boolean CanRead(
   System.Messaging.Message message
);
public final System.Object Clone();
public final System.Object Read(
   System.Messaging.Message message
);
public final void Write(
   System.Messaging.Message message,
   System.Object obj
);
```

CryptographicProviderType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CryptographicProviderType
```

Members

Dss

Fortezza

MicrosoftExchange

None

RsaFull

RsqSig

Ssl

SttAcq

SttBrnd

SttIss

SttMer

SttRoot

DefaultPropertiesToSend Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Compo
nentModel.ExpandableObjectConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyTo
ken=b77a5c561934e089") */
public class DefaultPropertiesToSend
   extends System.Object
```

Public Constructors

```
public DefaultPropertiesToSend();
```

Public Properties

```
public System.Messaging.AcknowledgeTypes get_AcknowledgeType();
public void set_AcknowledgeType(
   System.Messaging.AcknowledgeTypes value
);
public System.Messaging.MessageQueue get_AdministrationQueue();
public void set_AdministrationQueue(
   System.Messaging.MessageQueue value
);
public int get_AppSpecific();
public void set_AppSpecific(
   int value
);
public boolean get_AttachSenderId();
public void set_AttachSenderId(
   boolean value
);
public System.Messaging.EncryptionAlgorithm get_EncryptionAlgorithm();
public void set_EncryptionAlgorithm(
   System.Messaging.EncryptionAlgorithm value
);
public ubyte[] get_Extension();
public void set_Extension(
   ubyte[] value
);
public System.Messaging.HashAlgorithm get_HashAlgorithm();
public void set_HashAlgorithm(
   System.Messaging.HashAlgorithm value
);
public System.String get_Label();
public void set_Label(
   System.String value
);
public System.Messaging.MessagePriority get_Priority();
public void set_Priority(
   System.Messaging.MessagePriority value
);
public boolean get_Recoverable();
public void set_Recoverable(
   boolean value
);
public System.Messaging.MessageQueue get_ResponseQueue();
public void set_ResponseQueue(
   System.Messaging.MessageQueue value
```

```
);
public System.TimeSpan get_TimeToBeReceived();
public void set_TimeToBeReceived(
   System.TimeSpan value
);
public System.TimeSpan get_TimeToReachQueue();
public void set_TimeToReachQueue(
   System.TimeSpan value
);
public System.Messaging.MessageQueue get_TransactionStatusQueue();
public void set_TransactionStatusQueue(
   System.Messaging.MessageQueue value
);
public boolean get_UseAuthentication();
public void set_UseAuthentication(
   boolean value
);
public boolean get_UseDeadLetterQueue();
public void set_UseDeadLetterQueue(
   boolean value
);
public boolean get_UseEncryption();
public void set_UseEncryption(
   boolean value
);
public boolean get_UseJournalQueue();
public void set_UseJournalQueue(
   boolean value
);
public boolean get_UseTracing();
public void set_UseTracing(
   boolean value
);
```

EncryptionAlgorithm Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum EncryptionAlgorithm
```

Members

None

Rc2

Rc4

EncryptionRequired Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum EncryptionRequired
```

Members

Body

None

Optional

GenericAccessRights Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum GenericAccessRights
```

Members

All

Execute

None

Read

Write

HashAlgorithm Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum HashAlgorithm
```

Members

Mac

Md2

Md4

Md5

None

Sha

IMessageFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Messa
ging.Design.MessageFormatterConverter, System.Messaging, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a") */
public interface IMessageFormatter
   implements System.ICloneable
```

Public Methods

```
boolean CanRead(
   System.Messaging.Message message
);
System.Object Read(
   System.Messaging.Message message
);
void Write(
   System.Messaging.Message message,
   System.Object obj
);
```

Message Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Messaging.Design.MessageDesigner, Syste
m.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class Message
   extends System.ComponentModel.Component
```

Public Constructors

```
public Message();
public Message(
   System.Object body
);
public Message(
   System.Object body,
   System.Messaging.IMessageFormatter formatter
);
```

Public Properties

```
public System.Messaging.AcknowledgeTypes get_AcknowledgeType();
public void set_AcknowledgeType(
   System.Messaging.AcknowledgeTypes value
);
public System.Messaging.Acknowledgment get_Acknowledgment();
public System.Messaging.MessageQueue get_AdministrationQueue();
public void set_AdministrationQueue(
   System.Messaging.MessageQueue value
);
public int get_AppSpecific();
public void set_AppSpecific(
   int value
);
public System.DateTime get_ArrivedTime();
public boolean get_AttachSenderId();
public void set_AttachSenderId(
   boolean value
);
public boolean get_Authenticated();
public System.String get_AuthenticationProviderName();
public void set_AuthenticationProviderName(
   System.String value
);
public System.Messaging.CryptographicProviderType get_AuthenticationProviderType();
public void set_AuthenticationProviderType(
   System.Messaging.CryptographicProviderType value
);
public System.Object get_Body();
public void set_Body(
   System.Object value
);
public System.IO.Stream get_BodyStream();
public void set_BodyStream(
   System.IO.Stream value
);
public int get_BodyType();
```

```
public void set_BodyType(
   int value
);
public System.Guid get_ConnectorType();
public void set_ConnectorType(
   System.Guid value
);
public System.String get_CorrelationId();
public void set_CorrelationId(
   System.String value
);
public System.Messaging.MessageQueue get_DestinationQueue();
public ubyte[] get_DestinationSymmetricKey();
public void set_DestinationSymmetricKey(
   ubyte[] value
);
public ubyte[] get_DigitalSignature();
public void set_DigitalSignature(
   ubyte[] value
);
public System.Messaging.EncryptionAlgorithm get_EncryptionAlgorithm();
public void set_EncryptionAlgorithm(
   System.Messaging.EncryptionAlgorithm value
);
public ubyte[] get_Extension();
public void set_Extension(
   ubyte[] value
);
public System.Messaging.IMessageFormatter get_Formatter();
public void set_Formatter(
   System.Messaging.IMessageFormatter value
);
public System.Messaging.HashAlgorithm get_HashAlgorithm();
public void set_HashAlgorithm(
   System.Messaging.HashAlgorithm value
);
public System.String get_Id();
public boolean get_IsFirstInTransaction();
public boolean get_IsLastInTransaction();
public System.String get_Label();
public void set_Label(
   System.String value
);
public System.Messaging.MessageType get_MessageType();
public System.Messaging.MessagePriority get_Priority();
public void set_Priority(
   System.Messaging.MessagePriority value
);
public boolean get_Recoverable();
public void set_Recoverable(
   boolean value
);
public System.Messaging.MessageQueue get_ResponseQueue();
public void set_ResponseQueue(
   System.Messaging.MessageQueue value
);
public ubyte[] get_SenderCertificate();
public void set_SenderCertificate(
   ubyte[] value
);
public ubyte[] get_SenderId();
public long get_SenderVersion();
public System.DateTime get_SentTime();
public System.String get_SourceMachine();
public System.TimeSpan get_TimeToBeReceived();
public void set_TimeToBeReceived(
   System.TimeSpan value
);
```

```
public System.TimeSpan get_TimeToReachQueue();
public void set_TimeToReachQueue(
   System.TimeSpan value
);
public System.String get_TransactionId();
public System.Messaging.MessageQueue get_TransactionStatusQueue();
public void set_TransactionStatusQueue(
   System.Messaging.MessageQueue value
);
public boolean get_UseAuthentication();
public void set_UseAuthentication(
   boolean value
);
public boolean get_UseDeadLetterQueue();
public void set_UseDeadLetterQueue(
   boolean value
);
public boolean get_UseEncryption();
public void set_UseEncryption(
   boolean value
);
public boolean get_UseJournalQueue();
public void set_UseJournalQueue(
   boolean value
);
public boolean get_UseTracing();
public void set_UseTracing(
   boolean value
);
```

Public Fields

```
public static final System.TimeSpan InfiniteTimeout;
```

MessageEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MessageEnumerator
   extends System.MarshalByRefObject
   implements System.Collections.IEnumerator, System.IDisposable
```

Public Properties

```
public System.Messaging.Message get_Current();
public System.IntPtr get_CursorHandle();
```

Public Methods

```
public void Close();
public final void Dispose();
public final boolean MoveNext();
public boolean MoveNext(
   System.TimeSpan timeout
);
public System.Messaging.Message RemoveCurrent();
public System.Messaging.Message RemoveCurrent(
   System.Messaging.MessageQueueTransaction transaction
);
public System.Messaging.Message RemoveCurrent(
   System.Messaging.MessageQueueTransactionType transactionType
);
public System.Messaging.Message RemoveCurrent(
   System.TimeSpan timeout
);
public System.Messaging.Message RemoveCurrent(
   System.TimeSpan timeout,
   System.Messaging.MessageQueueTransaction transaction
);
public System.Messaging.Message RemoveCurrent(
   System.TimeSpan timeout,
   System.Messaging.MessageQueueTransactionType transactionType
);
public final void Reset();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

MessagePriority Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MessagePriority
```

Members

AboveNormal

High

Highest

Low

Lowest

Normal

VeryHigh

VeryLow

MessagePropertyFilter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Compo
nentModel.ExpandableObjectConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyTo
ken=b77a5c561934e089") */
public class MessagePropertyFilter
   extends System.Object
```

Public Constructors

```
public MessagePropertyFilter();
```

Public Properties

```
public boolean get_AcknowledgeType();
public void set_AcknowledgeType(
   boolean value
);
public boolean get_Acknowledgment();
public void set_Acknowledgment(
   boolean value
);
public boolean get_AdministrationQueue();
public void set_AdministrationQueue(
   boolean value
);
public boolean get_AppSpecific();
public void set_AppSpecific(
   boolean value
);
public boolean get_ArrivedTime();
public void set_ArrivedTime(
   boolean value
);
public boolean get_AttachSenderId();
public void set_AttachSenderId(
   boolean value
);
public boolean get_Authenticated();
public void set_Authenticated(
   boolean value
);
public boolean get_AuthenticationProviderName();
public void set_AuthenticationProviderName(
   boolean value
);
public boolean get_AuthenticationProviderType();
public void set_AuthenticationProviderType(
   boolean value
);
public boolean get_Body();
public void set_Body(
   boolean value
);
public boolean get_ConnectorType();
public void set_ConnectorType(
   boolean value
```

```
);
public boolean get_CorrelationId();
public void set_CorrelationId(
   boolean value
);
public int get_DefaultBodySize();
public void set_DefaultBodySize(
   int value
);
public int get_DefaultExtensionSize();
public void set_DefaultExtensionSize(
   int value
);
public int get_DefaultLabelSize();
public void set_DefaultLabelSize(
   int value
);
public boolean get_DestinationQueue();
public void set_DestinationQueue(
   boolean value
);
public boolean get_DestinationSymmetricKey();
public void set_DestinationSymmetricKey(
   boolean value
);
public boolean get_DigitalSignature();
public void set_DigitalSignature(
   boolean value
);
public boolean get_EncryptionAlgorithm();
public void set_EncryptionAlgorithm(
   boolean value
);
public boolean get_Extension();
public void set_Extension(
   boolean value
);
public boolean get_HashAlgorithm();
public void set_HashAlgorithm(
   boolean value
);
public boolean get_Id();
public void set_Id(
   boolean value
);
public boolean get_IsFirstInTransaction();
public void set_IsFirstInTransaction(
   boolean value
);
public boolean get_IsLastInTransaction();
public void set_IsLastInTransaction(
   boolean value
);
public boolean get_Label();
public void set_Label(
   boolean value
);
public boolean get_MessageType();
public void set_MessageType(
   boolean value
);
public boolean get_Priority();
public void set_Priority(
   boolean value
);
public boolean get_Recoverable();
public void set_Recoverable(
   boolean value
```

```
);
public boolean get_ResponseQueue();
public void set_ResponseQueue(
   boolean value
);
public boolean get_SenderCertificate();
public void set_SenderCertificate(
   boolean value
);
public boolean get_SenderId();
public void set_SenderId(
   boolean value
);
public boolean get_SenderVersion();
public void set_SenderVersion(
   boolean value
);
public boolean get_SentTime();
public void set_SentTime(
   boolean value
);
public boolean get_SourceMachine();
public void set_SourceMachine(
   boolean value
);
public boolean get_TimeToBeReceived();
public void set_TimeToBeReceived(
   boolean value
);
public boolean get_TimeToReachQueue();
public void set_TimeToReachQueue(
   boolean value
);
public boolean get_TransactionId();
public void set_TransactionId(
   boolean value
);
public boolean get_TransactionStatusQueue();
public void set_TransactionStatusQueue(
   boolean value
);
public boolean get_UseAuthentication();
public void set_UseAuthentication(
   boolean value
);
public boolean get_UseDeadLetterQueue();
public void set_UseDeadLetterQueue(
   boolean value
);
public boolean get_UseEncryption();
public void set_UseEncryption(
   boolean value
);
public boolean get_UseJournalQueue();
public void set_UseJournalQueue(
   boolean value
);
public boolean get_UseTracing();
public void set_UseTracing(
   boolean value
);
```

Public Methods

```
public void ClearAll();
public void SetAll();
```

```
public void SetDefaults();
```

MessageQueue Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Messa
ging.Design.MessageQueueConverter, System.Messaging, Version=1.0.3300.0, Culture=neutral, Pub
licKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Messaging.Design.QueuePathEditor") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "Microsoft.VisualStudio.Install.MessageQueueIns
tallableComponentDesigner, Microsoft.VisualStudio, Version=1.0.3300.0, Culture=neutral, Publi
cKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.InstallerTypeAttribute(InstallerType = System.Messaging.
MessageQueueInstaller) */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ReceiveCompleted") */
public class MessageQueue
   extends System.ComponentModel.Component
   implements System.Collections.IEnumerable
```

Public Constructors

```
public MessageQueue();
public MessageQueue(
   System.String path
);
public MessageQueue(
   System.String path,
   boolean sharedModeDenyReceive
);
```

Public Properties

```
public boolean get_Authenticate();
public void set_Authenticate(
   boolean value
);
public short get_BasePriority();
public void set_BasePriority(
   short value
);
public boolean get_CanRead();
public boolean get_CanWrite();
public System.Guid get_Category();
public void set_Category(
   System.Guid value
);
public System.DateTime get_CreateTime();
public System.Messaging.DefaultPropertiesToSend get_DefaultPropertiesToSend();
public void set_DefaultPropertiesToSend(
   System.Messaging.DefaultPropertiesToSend value
);
public boolean get_DenySharedReceive();
public void set_DenySharedReceive(
   boolean value
);
public static boolean get_EnableConnectionCache();
public static void set_EnableConnectionCache(
```

```
   boolean value
);
public System.Messaging.EncryptionRequired get_EncryptionRequired();
public void set_EncryptionRequired(
   System.Messaging.EncryptionRequired value
);
public System.String get_FormatName();
public System.Messaging.IMessageFormatter get_Formatter();
public void set_Formatter(
   System.Messaging.IMessageFormatter value
);
public System.Guid get_Id();
public System.String get_Label();
public void set_Label(
   System.String value
);
public System.DateTime get_LastModifyTime();
public System.String get_MachineName();
public void set_MachineName(
   System.String value
);
public long get_MaximumJournalSize();
public void set_MaximumJournalSize(
   long value
);
public long get_MaximumQueueSize();
public void set_MaximumQueueSize(
   long value
);
public System.Messaging.MessagePropertyFilter get_MessageReadPropertyFilter();
public void set_MessageReadPropertyFilter(
   System.Messaging.MessagePropertyFilter value
);
public System.String get_Path();
public void set_Path(
   System.String value
);
public System.String get_QueueName();
public void set_QueueName(
   System.String value
);
public System.IntPtr get_ReadHandle();
public System.ComponentModel.ISynchronizeInvoke get_SynchronizingObject();
public void set_SynchronizingObject(
   System.ComponentModel.ISynchronizeInvoke value
);
public boolean get_Transactional();
public boolean get_UseJournalQueue();
public void set_UseJournalQueue(
   boolean value
);
public System.IntPtr get_WriteHandle();
```

Public Methods

```
public System.IAsyncResult BeginPeek();
public System.IAsyncResult BeginPeek(
   System.TimeSpan timeout
);
public System.IAsyncResult BeginPeek(
   System.TimeSpan timeout,
   System.Object stateObject
);
public System.IAsyncResult BeginPeek(
   System.TimeSpan timeout,
   System.Object stateObject,
```

```
   System.AsyncCallback callback
);
public System.IAsyncResult BeginReceive();
public System.IAsyncResult BeginReceive(
   System.TimeSpan timeout
);
public System.IAsyncResult BeginReceive(
   System.TimeSpan timeout,
   System.Object stateObject
);
public System.IAsyncResult BeginReceive(
   System.TimeSpan timeout,
   System.Object stateObject,
   System.AsyncCallback callback
);
public static void ClearConnectionCache();
public void Close();
public static System.Messaging.MessageQueue Create(
   System.String path
);
public static System.Messaging.MessageQueue Create(
   System.String path,
   boolean transactional
);
public static void Delete(
   System.String path
);
public System.Messaging.Message EndPeek(
   System.IAsyncResult asyncResult
);
public System.Messaging.Message EndReceive(
   System.IAsyncResult asyncResult
);
public static boolean Exists(
   System.String path
);
public System.Messaging.Message[] GetAllMessages();
public final System.Collections.IEnumerator GetEnumerator();
public static System.Guid GetMachineId(
   System.String machineName
);
public System.Messaging.MessageEnumerator GetMessageEnumerator();
public static System.Messaging.MessageQueueEnumerator GetMessageQueueEnumerator();
public static System.Messaging.MessageQueueEnumerator GetMessageQueueEnumerator(
   System.Messaging.MessageQueueCriteria criteria
);
public static System.Messaging.MessageQueue[] GetPrivateQueuesByMachine(
   System.String machineName
);
public static System.Messaging.MessageQueue[] GetPublicQueues();
public static System.Messaging.MessageQueue[] GetPublicQueues(
   System.Messaging.MessageQueueCriteria criteria
);
public static System.Messaging.MessageQueue[] GetPublicQueuesByCategory(
   System.Guid category
);
public static System.Messaging.MessageQueue[] GetPublicQueuesByLabel(
   System.String label
);
public static System.Messaging.MessageQueue[] GetPublicQueuesByMachine(
   System.String machineName
);
public System.Messaging.Message Peek();
public System.Messaging.Message Peek(
   System.TimeSpan timeout
);
public System.Messaging.Message PeekByCorrelationId(
   System.String correlationId
```

```
);
public System.Messaging.Message PeekByCorrelationId(
   System.String correlationId,
   System.TimeSpan timeout
);
public System.Messaging.Message PeekById(
   System.String id
);
public System.Messaging.Message PeekById(
   System.String id,
   System.TimeSpan timeout
);
public void Purge();
public System.Messaging.Message Receive();
public System.Messaging.Message Receive(
   System.Messaging.MessageQueueTransaction transaction
);
public System.Messaging.Message Receive(
   System.Messaging.MessageQueueTransactionType transactionType
);
public System.Messaging.Message Receive(
   System.TimeSpan timeout
);
public System.Messaging.Message Receive(
   System.TimeSpan timeout,
   System.Messaging.MessageQueueTransaction transaction
);
public System.Messaging.Message Receive(
   System.TimeSpan timeout,
   System.Messaging.MessageQueueTransactionType transactionType
);
public System.Messaging.Message ReceiveByCorrelationId(
   System.String correlationId
);
public System.Messaging.Message ReceiveByCorrelationId(
   System.String correlationId,
   System.Messaging.MessageQueueTransaction transaction
);
public System.Messaging.Message ReceiveByCorrelationId(
   System.String correlationId,
   System.Messaging.MessageQueueTransactionType transactionType
);
public System.Messaging.Message ReceiveByCorrelationId(
   System.String correlationId,
   System.TimeSpan timeout
);
public System.Messaging.Message ReceiveByCorrelationId(
   System.String correlationId,
   System.TimeSpan timeout,
   System.Messaging.MessageQueueTransaction transaction
);
public System.Messaging.Message ReceiveByCorrelationId(
   System.String correlationId,
   System.TimeSpan timeout,
   System.Messaging.MessageQueueTransactionType transactionType
);
public System.Messaging.Message ReceiveById(
   System.String id
);
public System.Messaging.Message ReceiveById(
   System.String id,
   System.Messaging.MessageQueueTransaction transaction
);
public System.Messaging.Message ReceiveById(
   System.String id,
   System.Messaging.MessageQueueTransactionType transactionType
);
public System.Messaging.Message ReceiveById(
```

```
   System.String id,
   System.TimeSpan timeout
);
public System.Messaging.Message ReceiveById(
   System.String id,
   System.TimeSpan timeout,
   System.Messaging.MessageQueueTransaction transaction
);
public System.Messaging.Message ReceiveById(
   System.String id,
   System.TimeSpan timeout,
   System.Messaging.MessageQueueTransactionType transactionType
);
public void Refresh();
public void ResetPermissions();
public void Send(
   System.Object obj
);
public void Send(
   System.Object obj,
   System.Messaging.MessageQueueTransaction transaction
);
public void Send(
   System.Object obj,
   System.Messaging.MessageQueueTransactionType transactionType
);
public void Send(
   System.Object obj,
   System.String label
);
public void Send(
   System.Object obj,
   System.String label,
   System.Messaging.MessageQueueTransaction transaction
);
public void Send(
   System.Object obj,
   System.String label,
   System.Messaging.MessageQueueTransactionType transactionType
);
public void SetPermissions(
   System.Messaging.AccessControlList dacl
);
public void SetPermissions(
   System.Messaging.MessageQueueAccessControlEntry ace
);
public void SetPermissions(
   System.String user,
   System.Messaging.MessageQueueAccessRights rights
);
public void SetPermissions(
   System.String user,
   System.Messaging.MessageQueueAccessRights rights,
   System.Messaging.AccessControlEntryType entryType
);
```

Public Events

```
public void add_PeekCompleted(
   System.Messaging.PeekCompletedEventHandler value
);
public void remove_PeekCompleted(
   System.Messaging.PeekCompletedEventHandler value
);
public void add_ReceiveCompleted(
   System.Messaging.ReceiveCompletedEventHandler value
```

```
);
public void remove_ReceiveCompleted(
   System.Messaging.ReceiveCompletedEventHandler value
);
```

Public Fields

```
public static final long InfiniteQueueSize;
public static final System.TimeSpan InfiniteTimeout;
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

MessageQueueAccessControlEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MessageQueueAccessControlEntry
   extends System.Messaging.AccessControlEntry
```

Public Constructors

```
public MessageQueueAccessControlEntry(
   System.Messaging.Trustee trustee,
   System.Messaging.MessageQueueAccessRights rights
);
public MessageQueueAccessControlEntry(
   System.Messaging.Trustee trustee,
   System.Messaging.MessageQueueAccessRights rights,
   System.Messaging.AccessControlEntryType entryType
);
```

Public Properties

```
public System.Messaging.MessageQueueAccessRights get_MessageQueueAccessRights();
public void set_MessageQueueAccessRights(
   System.Messaging.MessageQueueAccessRights value
);
```

MessageQueueAccessRights Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum MessageQueueAccessRights
```

Members

ChangeQueuePermissions

DeleteJournalMessage

DeleteMessage

DeleteQueue

FullControl

GenericRead

GenericWrite

GetQueuePermissions

GetQueueProperties

PeekMessage

ReceiveJournalMessage

ReceiveMessage

SetQueueProperties

TakeQueueOwnership

WriteMessage

MessageQueueCriteria Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MessageQueueCriteria
   extends System.Object
```

Public Constructors

```
public MessageQueueCriteria();
```

Public Properties

```
public System.Guid get_Category();
public void set_Category(
   System.Guid value
);
public System.DateTime get_CreatedAfter();
public void set_CreatedAfter(
   System.DateTime value
);
public System.DateTime get_CreatedBefore();
public void set_CreatedBefore(
   System.DateTime value
);
public System.String get_Label();
public void set_Label(
   System.String value
);
public System.String get_MachineName();
public void set_MachineName(
   System.String value
);
public System.DateTime get_ModifiedAfter();
public void set_ModifiedAfter(
   System.DateTime value
);
public System.DateTime get_ModifiedBefore();
public void set_ModifiedBefore(
   System.DateTime value
);
```

Public Methods

```
public void ClearAll();
```

MessageQueueEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MessageQueueEnumerator
   extends System.MarshalByRefObject
   implements System.Collections.IEnumerator, System.IDisposable
```

Public Properties

```
public System.Messaging.MessageQueue get_Current();
public System.IntPtr get_LocatorHandle();
```

Public Methods

```
public void Close();
public final void Dispose();
public final boolean MoveNext();
public final void Reset();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

MessageQueueErrorCode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MessageQueueErrorCode
```

Members

AccessDenied

BadSecurityContext

Base

BufferOverflow

CannotCreateCertificateStore

CannotCreateHashEx

CannotCreateOnGlobalCatalog

CannotGetDistinguishedName

CannotGrantAddGuid

CannotHashDataEx

CannotImpersonateClient

CannotJoinDomain

CannotLoadMsmqOcm

CannotOpenCertificateStore

CannotSetCryptographicSecurityDescriptor

CannotSignDataEx

CertificateNotProvided

ComputerDoesNotSupportEncryption

CorruptedInternalCertificate

CorruptedPersonalCertStore

CorruptedQueueWasDeleted

CorruptedSecurityData

CouldNotGetAccountInfo

CouldNotGetUserSid

DeleteConnectedNetworkInUse

DependentClientLicenseOverflow

DsError

DsIsFull

DtcConnect

EncryptionProviderNotSupported

FailVerifySignatureEx

FormatNameBufferTooSmall

Generic

GuidNotMatching

IllegalContext

IllegalCriteriaColumns

IllegalCursorAction

IllegalEnterpriseOperation

IllegalFormatName

IllegalMessageProperties

IllegalOperation

IllegalPrivateProperties

IllegalPropertyId

IllegalPropertySize

IllegalPropertyValue

IllegalPropertyVt

IllegalQueuePathName

IllegalQueueProperties

IllegalRelation

IllegalRestrictionPropertyId

IllegalSecurityDescriptor

IllegalSort

IllegalSortPropertyId

IllegalUser

InsufficientProperties

InsufficientResources

InvalidCertificate

InvalidHandle

InvalidOwner

InvalidParameter

IOTimeout

LabelBufferTooSmall

MachineExists

MachineNotFound

MessageAlreadyReceived

MessageStorageFailed

MissingConnectorType

MqisReadOnlyMode

MqisServerEmpty

NoDs

NoEntryPointMsmqOcm

NoGlobalCatalogInDomain

NoInternalUserCertificate

NoMsmqServersOnDc

NoMsmqServersOnGlobalCatalog

NoResponseFromObjectServer

ObjectServerNotAvailable

OperationCanceled

PrivilegeNotHeld

Property

PropertyNotAllowed

ProviderNameBufferTooSmall

PublicKeyDoesNotExist

PublicKeyNotFound

QDnsPropertyNotSupported

QueueDeleted

QueueExists

QueueNotAvailable

QueueNotFound

RemoteMachineNotAvailable

ResultBufferTooSmall

SecurityDescriptorBufferTooSmall

SenderCertificateBufferTooSmall

SenderIdBufferTooSmall

ServiceNotAvailable

SharingViolation

SignatureBufferTooSmall

StaleHandle

SymmetricKeyBufferTooSmall

TransactionEnlist

TransactionImport

TransactionSequence

TransactionUsage

UnsupportedAccessMode

UnsupportedFormatNameOperation

UnsupportedOperation

UserBufferTooSmall

WksCantServeClient

WriteNotAllowed

MessageQueueException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class MessageQueueException
   extends System.Runtime.InteropServices.ExternalException
```

Public Properties

```
public System.String get_Message();
public System.Messaging.MessageQueueErrorCode get_MessageQueueErrorCode();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected MessageQueueException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

MessageQueueInstaller Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MessageQueueInstaller
   extends System.Configuration.Install.ComponentInstaller
```

Public Constructors

```
public MessageQueueInstaller();
public MessageQueueInstaller(
   System.Messaging.MessageQueue componentToCopy
);
```

Public Properties

```
public boolean get_Authenticate();
public void set_Authenticate(
   boolean value
);
public short get_BasePriority();
public void set_BasePriority(
   short value
);
public System.Guid get_Category();
public void set_Category(
   System.Guid value
);
public System.Messaging.EncryptionRequired get_EncryptionRequired();
public void set_EncryptionRequired(
   System.Messaging.EncryptionRequired value
);
public System.String get_Label();
public void set_Label(
   System.String value
);
public long get_MaximumJournalSize();
public void set_MaximumJournalSize(
   long value
);
public long get_MaximumQueueSize();
public void set_MaximumQueueSize(
   long value
);
public System.String get_Path();
public void set_Path(
   System.String value
);
public System.Messaging.AccessControlList get_Permissions();
public void set_Permissions(
   System.Messaging.AccessControlList value
);
public boolean get_Transactional();
public void set_Transactional(
   boolean value
);
public System.Configuration.Install.UninstallAction get_UninstallAction();
public void set_UninstallAction(
   System.Configuration.Install.UninstallAction value
```

```
);
public boolean get_UseJournalQueue();
public void set_UseJournalQueue(
   boolean value
);
```

Public Methods

```
public void Commit(
   System.Collections.IDictionary savedState
);
public void CopyFromComponent(
   System.ComponentModel.IComponent component
);
public void Install(
   System.Collections.IDictionary stateSaver
);
public boolean IsEquivalentInstaller(
   System.Configuration.Install.ComponentInstaller otherInstaller
);
public void Rollback(
   System.Collections.IDictionary savedState
);
public void Uninstall(
   System.Collections.IDictionary savedState
);
```

MessageQueuePermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class MessageQueuePermission
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Constructors

```
public MessageQueuePermission();
public MessageQueuePermission(
   System.Messaging.MessageQueuePermissionEntry[] permissionAccessEntries
);
public MessageQueuePermission(
   System.Security.Permissions.PermissionState state
);
public MessageQueuePermission(
   System.Messaging.MessageQueuePermissionAccess permissionAccess,
   System.String path
);
public MessageQueuePermission(
   System.Messaging.MessageQueuePermissionAccess permissionAccess,
   System.String machineName,
   System.String label,
   System.String category
);
```

Public Properties

```
public System.Messaging.MessageQueuePermissionEntryCollection get_PermissionEntries();
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement securityElement
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public final boolean IsUnrestricted();
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

MessageQueuePermissionAccess Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum MessageQueuePermissionAccess
```

Members

Administer

Browse

None

Peek

Receive

Send

MessageQueuePermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, Event, AllowMultiple = true, Inherited = false) */
public class MessageQueuePermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public MessageQueuePermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.String get_Category();
public void set_Category(
   System.String value
);
public System.String get_Label();
public void set_Label(
   System.String value
);
public System.String get_MachineName();
public void set_MachineName(
   System.String value
);
public System.String get_Path();
public void set_Path(
   System.String value
);
public System.Messaging.MessageQueuePermissionAccess get_PermissionAccess();
public void set_PermissionAccess(
   System.Messaging.MessageQueuePermissionAccess value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

MessageQueuePermissionEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class MessageQueuePermissionEntry
   extends System.Object
```

Public Constructors

```
public MessageQueuePermissionEntry(
   System.Messaging.MessageQueuePermissionAccess permissionAccess,
   System.String path
);
public MessageQueuePermissionEntry(
   System.Messaging.MessageQueuePermissionAccess permissionAccess,
   System.String machineName,
   System.String label,
   System.String category
);
```

Public Properties

```
public System.String get_Category();
public System.String get_Label();
public System.String get_MachineName();
public System.String get_Path();
public System.Messaging.MessageQueuePermissionAccess get_PermissionAccess();
```

MessageQueuePermissionEntryCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class MessageQueuePermissionEntryCollection
   extends System.Collections.CollectionBase
```

Public Properties

```
public System.Messaging.MessageQueuePermissionEntry get_Item(
   int index
);
public void set_Item(
   int index,
   System.Messaging.MessageQueuePermissionEntry value
);
```

Public Methods

```
public int Add(
   System.Messaging.MessageQueuePermissionEntry value
);
public void AddRange(
   System.Messaging.MessageQueuePermissionEntryCollection value
);
public void AddRange(
   System.Messaging.MessageQueuePermissionEntry[] value
);
public boolean Contains(
   System.Messaging.MessageQueuePermissionEntry value
);
public void CopyTo(
   System.Messaging.MessageQueuePermissionEntry[] array,
   int index
);
public int IndexOf(
   System.Messaging.MessageQueuePermissionEntry value
);
public void Insert(
   int index,
   System.Messaging.MessageQueuePermissionEntry value
);
public void Remove(
   System.Messaging.MessageQueuePermissionEntry value
);
```

Protected Methods

```
protected void OnClear();
protected void OnInsert(
   int index,
   System.Object value
```

```
);
protected void OnRemove(
   int index,
   System.Object value
);
protected void OnSet(
   int index,
   System.Object oldValue,
   System.Object newValue
);
```

MessageQueueTransaction Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MessageQueueTransaction
   extends System.Object
   implements System.IDisposable
```

Public Constructors

```
public MessageQueueTransaction();
```

Public Properties

```
public System.Messaging.MessageQueueTransactionStatus get_Status();
```

Public Methods

```
public void Abort();
public void Begin();
public void Commit();
public final void Dispose();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

MessageQueueTransactionStatus Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MessageQueueTransactionStatus
```

Members

Aborted

Committed

Initialized

Pending

MessageQueueTransactionType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MessageQueueTransactionType
```

Members

Automatic

None

Single

MessageType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MessageType
```

Members

Acknowledgment

Normal

Report

MessagingDescriptionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class MessagingDescriptionAttribute
   extends System.ComponentModel.DescriptionAttribute
```

Public Constructors

```
public MessagingDescriptionAttribute(
   System.String description
);
```

Public Properties

```
public System.String get_Description();
```

PeekCompletedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PeekCompletedEventArgs
   extends System.EventArgs
```

Public Properties

```
public System.IAsyncResult get_AsyncResult();
public void set_AsyncResult(
   System.IAsyncResult value
);
public System.Messaging.Message get_Message();
```

PeekCompletedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void PeekCompletedEventHandler(
   System.Object sender,
   System.Messaging.PeekCompletedEventArgs e
);
```

ReceiveCompletedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ReceiveCompletedEventArgs
   extends System.EventArgs
```

Public Properties

```
public System.IAsyncResult get_AsyncResult();
public void set_AsyncResult(
   System.IAsyncResult value
);
public System.Messaging.Message get_Message();
```

ReceiveCompletedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ReceiveCompletedEventHandler(
   System.Object sender,
   System.Messaging.ReceiveCompletedEventArgs e
);
```

StandardAccessRights Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum StandardAccessRights
```

Members

All

Delete

Execute

ModifyOwner

None

Read

ReadSecurity

Required

Synchronize

Write

WriteSecurity

Trustee Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Trustee
   extends System.Object
```

Public Constructors

```
public Trustee();
public Trustee(
   System.String name
);
public Trustee(
   System.String name,
   System.String systemName
);
public Trustee(
   System.String name,
   System.String systemName,
   System.Messaging.TrusteeType trusteeType
);
```

Public Properties

```
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_SystemName();
public void set_SystemName(
   System.String value
);
public System.Messaging.TrusteeType get_TrusteeType();
public void set_TrusteeType(
   System.Messaging.TrusteeType value
);
```

TrusteeType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TrusteeType
```

Members

Alias

Computer

Domain

Group

Unknown

User

XmlMessageFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlMessageFormatter
   extends System.Object
   implements System.Messaging.IMessageFormatter, System.ICloneable
```

Public Constructors

```
public XmlMessageFormatter();
public XmlMessageFormatter(
   System.String[] targetTypeNames
);
public XmlMessageFormatter(
   System.Type[] targetTypes
);
```

Public Properties

```
public System.String[] get_TargetTypeNames();
public void set_TargetTypeNames(
   System.String[] value
);
public System.Type[] get_TargetTypes();
public void set_TargetTypes(
   System.Type[] value
);
```

Public Methods

```
public final boolean CanRead(
   System.Messaging.Message message
);
public final System.Object Clone();
public final System.Object Read(
   System.Messaging.Message message
);
public final void Write(
   System.Messaging.Message message,
   System.Object obj
);
```

System.Net Namespace (Visual J# Syntax)

Classes

AuthenticationManager

Authorization

Cookie

CookieCollection

CookieContainer

CookieException

CredentialCache

Dns

DnsPermission

DnsPermissionAttribute

EndPoint

EndpointPermission

FileWebRequest

FileWebResponse

GlobalProxySelection

HttpVersion

HttpWebRequest

HttpWebResponse

IPAddress

IPEndPoint

IPHostEntry

NetworkCredential

ProtocolViolationException

ServicePoint

ServicePointManager

SocketAddress

SocketPermission

SocketPermissionAttribute

WebClient

WebException

WebHeaderCollection

WebPermission

WebPermissionAttribute

WebProxy

WebRequest

WebResponse

Interfaces

IAuthenticationModule

ICertificatePolicy

ICredentials

IWebProxy

IWebRequestCreate

Delegates

HttpContinueDelegate

Enumerations

HttpStatusCode

NetworkAccess

TransportType

WebExceptionStatus

AuthenticationManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AuthenticationManager
   extends System.Object
```

Public Properties

```
public static System.Collections.IEnumerator get_RegisteredModules();
```

Public Methods

```
public static System.Net.Authorization Authenticate(
   System.String challenge,
   System.Net.WebRequest request,
   System.Net.ICredentials credentials
);
public static System.Net.Authorization PreAuthenticate(
   System.Net.WebRequest request,
   System.Net.ICredentials credentials
);
public static void Register(
   System.Net.IAuthenticationModule authenticationModule
);
public static void Unregister(
   System.Net.IAuthenticationModule authenticationModule
);
public static void Unregister(
   System.String authenticationScheme
);
```

Authorization Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Authorization
   extends System.Object
```

Public Constructors

```
public Authorization(
   System.String token
);
public Authorization(
   System.String token,
   boolean finished
);
public Authorization(
   System.String token,
   boolean finished,
   System.String connectionGroupId
);
```

Public Properties

```
public boolean get_Complete();
public System.String get_ConnectionGroupId();
public System.String get_Message();
public System.String[] get_ProtectionRealm();
public void set_ProtectionRealm(
   System.String[] value
);
```

Cookie Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Cookie
   extends System.Object
```

Public Constructors

```
public Cookie();
public Cookie(
   System.String name,
   System.String value
);
public Cookie(
   System.String name,
   System.String value,
   System.String path
);
public Cookie(
   System.String name,
   System.String value,
   System.String path,
   System.String domain
);
```

Public Properties

```
public System.String get_Comment();
public void set_Comment(
   System.String value
);
public System.Uri get_CommentUri();
public void set_CommentUri(
   System.Uri value
);
public boolean get_Discard();
public void set_Discard(
   boolean value
);
public System.String get_Domain();
public void set_Domain(
   System.String value
);
public boolean get_Expired();
public void set_Expired(
   boolean value
);
public System.DateTime get_Expires();
public void set_Expires(
   System.DateTime value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_Path();
public void set_Path(
```

```
   System.String value
);
public System.String get_Port();
public void set_Port(
   System.String value
);
public boolean get_Secure();
public void set_Secure(
   boolean value
);
public System.DateTime get_TimeStamp();
public System.String get_Value();
public void set_Value(
   System.String value
);
public int get_Version();
public void set_Version(
   int value
);
```

Public Methods

```
public boolean Equals(
   System.Object comparand
);
public int GetHashCode();
public System.String ToString();
```

CookieCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CookieCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public CookieCollection();
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Net.Cookie get_Item(
   int index
);
public System.Net.Cookie get_Item(
   System.String name
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public void Add(
   System.Net.Cookie cookie
);
public void Add(
   System.Net.CookieCollection cookies
);
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
```

CookieContainer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class CookieContainer
   extends System.Object
```

Public Constructors

```
public CookieContainer();
public CookieContainer(
   int capacity
);
public CookieContainer(
   int capacity,
   int perDomainCapacity,
   int maxCookieSize
);
```

Public Properties

```
public int get_Capacity();
public void set_Capacity(
   int value
);
public int get_Count();
public int get_MaxCookieSize();
public void set_MaxCookieSize(
   int value
);
public int get_PerDomainCapacity();
public void set_PerDomainCapacity(
   int value
);
```

Public Methods

```
public void Add(
   System.Net.Cookie cookie
);
public void Add(
   System.Net.CookieCollection cookies
);
public void Add(
   System.Uri uri,
   System.Net.Cookie cookie
);
public void Add(
   System.Uri uri,
   System.Net.CookieCollection cookies
);
public System.String GetCookieHeader(
   System.Uri uri
);
public System.Net.CookieCollection GetCookies(
   System.Uri uri
```

```
);
public void SetCookies(
   System.Uri uri,
   System.String cookieHeader
);
```

Public Fields

```
public static int DefaultCookieLengthLimit;
public static int DefaultCookieLimit;
public static int DefaultPerDomainCookieLimit;
```

CookieException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class CookieException
   extends System.FormatException
```

Public Constructors

```
public CookieException();
```

Protected Constructors

```
protected CookieException(
   System.Runtime.Serialization.SerializationInfo serializationInfo,
   System.Runtime.Serialization.StreamingContext streamingContext
);
```

CredentialCache Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CredentialCache
   extends System.Object
   implements System.Net.ICredentials, System.Collections.IEnumerable
```

Public Constructors

```
public CredentialCache();
```

Public Properties

```
public static System.Net.ICredentials get_DefaultCredentials();
```

Public Methods

```
public void Add(
   System.Uri uriPrefix,
   System.String authType,
   System.Net.NetworkCredential cred
);
public final System.Net.NetworkCredential GetCredential(
   System.Uri uriPrefix,
   System.String authType
);
public final System.Collections.IEnumerator GetEnumerator();
public void Remove(
   System.Uri uriPrefix,
   System.String authType
);
```

Dns Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Dns
   extends System.Object
```

Public Methods

```
public static System.IAsyncResult BeginGetHostByName(
   System.String hostName,
   System.AsyncCallback requestCallback,
   System.Object stateObject
);
public static System.IAsyncResult BeginResolve(
   System.String hostName,
   System.AsyncCallback requestCallback,
   System.Object stateObject
);
public static System.Net.IPHostEntry EndGetHostByName(
   System.IAsyncResult asyncResult
);
public static System.Net.IPHostEntry EndResolve(
   System.IAsyncResult asyncResult
);
public static System.Net.IPHostEntry GetHostByAddress(
   System.Net.IPAddress address
);
public static System.Net.IPHostEntry GetHostByAddress(
   System.String address
);
public static System.Net.IPHostEntry GetHostByName(
   System.String hostName
);
public static System.String GetHostName();
public static System.Net.IPHostEntry Resolve(
   System.String hostName
);
```

DnsPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DnsPermission
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Constructors

```
public DnsPermission(
   System.Security.Permissions.PermissionState state
);
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement securityElement
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public final boolean IsUnrestricted();
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

DnsPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class DnsPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public DnsPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

EndPoint Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class EndPoint
   extends System.Object
```

Public Properties

```
public System.Net.Sockets.AddressFamily get_AddressFamily();
```

Public Methods

```
public System.Net.EndPoint Create(
   System.Net.SocketAddress socketAddress
);
public System.Net.SocketAddress Serialize();
```

Protected Constructors

```
protected EndPoint();
```

EndpointPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class EndpointPermission
   extends System.Object
```

Public Properties

```
public System.String get_Hostname();
public int get_Port();
public System.Net.TransportType get_Transport();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public System.String ToString();
```

FileWebRequest Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class FileWebRequest
   extends System.Net.WebRequest
```

Public Properties

```
public System.String get_ConnectionGroupName();
public void set_ConnectionGroupName(
   System.String value
);
public long get_ContentLength();
public void set_ContentLength(
   long value
);
public System.String get_ContentType();
public void set_ContentType(
   System.String value
);
public System.Net.ICredentials get_Credentials();
public void set_Credentials(
   System.Net.ICredentials value
);
public System.Net.WebHeaderCollection get_Headers();
public System.String get_Method();
public void set_Method(
   System.String value
);
public boolean get_PreAuthenticate();
public void set_PreAuthenticate(
   boolean value
);
public System.Net.IWebProxy get_Proxy();
public void set_Proxy(
   System.Net.IWebProxy value
);
public System.Uri get_RequestUri();
public int get_Timeout();
public void set_Timeout(
   int value
);
```

Public Methods

```
public System.IAsyncResult BeginGetRequestStream(
   System.AsyncCallback callback,
   System.Object state
);
public System.IAsyncResult BeginGetResponse(
   System.AsyncCallback callback,
   System.Object state
);
public System.IO.Stream EndGetRequestStream(
   System.IAsyncResult asyncResult
);
public System.Net.WebResponse EndGetResponse(
```

```
   System.IAsyncResult asyncResult
);
public System.IO.Stream GetRequestStream();
public System.Net.WebResponse GetResponse();
```

Protected Constructors

```
protected FileWebRequest(
   System.Runtime.Serialization.SerializationInfo serializationInfo,
   System.Runtime.Serialization.StreamingContext streamingContext
);
```

FileWebResponse Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class FileWebResponse
   extends System.Net.WebResponse
```

Public Properties

```
public long get_ContentLength();
public System.String get_ContentType();
public System.Net.WebHeaderCollection get_Headers();
public System.Uri get_ResponseUri();
```

Public Methods

```
public void Close();
public System.IO.Stream GetResponseStream();
```

Protected Constructors

```
protected FileWebResponse(
   System.Runtime.Serialization.SerializationInfo serializationInfo,
   System.Runtime.Serialization.StreamingContext streamingContext
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

GlobalProxySelection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class GlobalProxySelection
   extends System.Object
```

Public Constructors

```
public GlobalProxySelection();
```

Public Properties

```
public static System.Net.IWebProxy get_Select();
public static void set_Select(
   System.Net.IWebProxy value
);
```

Public Methods

```
public static System.Net.IWebProxy GetEmptyWebProxy();
```

HttpContinueDelegate Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void HttpContinueDelegate(
   int StatusCode,
   System.Net.WebHeaderCollection httpHeaders
);
```

HttpStatusCode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum HttpStatusCode
```

Members

Accepted

Ambiguous

BadGateway

BadRequest

Conflict

Continue

Created

ExpectationFailed

Forbidden

Found

GatewayTimeout

Gone

HttpVersionNotSupported

InternalServerError

LengthRequired

MethodNotAllowed

Moved

MovedPermanently

MultipleChoices

NoContent

NonAuthoritativeInformation

NotAcceptable

NotFound

NotImplemented

NotModified

OK

PartialContent

PaymentRequired

PreconditionFailed

ProxyAuthenticationRequired

Redirect

RedirectKeepVerb

RedirectMethod

RequestedRangeNotSatisfiable

RequestEntityTooLarge

RequestTimeout

RequestUriTooLong

ResetContent

SeeOther

ServiceUnavailable

SwitchingProtocols

TemporaryRedirect

Unauthorized

UnsupportedMediaType

Unused

UseProxy

HttpVersion Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpVersion
   extends System.Object
```

Public Constructors

```
public HttpVersion();
```

Public Fields

```
public static final System.Version Version10;
public static final System.Version Version11;
```

HttpWebRequest Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class HttpWebRequest
   extends System.Net.WebRequest
```

Public Properties

```
public System.String get_Accept();
public void set_Accept(
   System.String value
);
public System.Uri get_Address();
public boolean get_AllowAutoRedirect();
public void set_AllowAutoRedirect(
   boolean value
);
public boolean get_AllowWriteStreamBuffering();
public void set_AllowWriteStreamBuffering(
   boolean value
);
public System.Security.Cryptography.X509Certificates.X509CertificateCollection get_ClientCert
ificates();
public System.String get_Connection();
public void set_Connection(
   System.String value
);
public System.String get_ConnectionGroupName();
public void set_ConnectionGroupName(
   System.String value
);
public long get_ContentLength();
public void set_ContentLength(
   long value
);
public System.String get_ContentType();
public void set_ContentType(
   System.String value
);
public System.Net.HttpContinueDelegate get_ContinueDelegate();
public void set_ContinueDelegate(
   System.Net.HttpContinueDelegate value
);
public System.Net.CookieContainer get_CookieContainer();
public void set_CookieContainer(
   System.Net.CookieContainer value
);
public System.Net.ICredentials get_Credentials();
public void set_Credentials(
   System.Net.ICredentials value
);
public System.String get_Expect();
public void set_Expect(
   System.String value
);
public boolean get_HaveResponse();
public System.Net.WebHeaderCollection get_Headers();
public void set_Headers(
   System.Net.WebHeaderCollection value
```

```
);
public System.DateTime get_IfModifiedSince();
public void set_IfModifiedSince(
   System.DateTime value
);
public boolean get_KeepAlive();
public void set_KeepAlive(
   boolean value
);
public int get_MaximumAutomaticRedirections();
public void set_MaximumAutomaticRedirections(
   int value
);
public System.String get_MediaType();
public void set_MediaType(
   System.String value
);
public System.String get_Method();
public void set_Method(
   System.String value
);
public boolean get_Pipelined();
public void set_Pipelined(
   boolean value
);
public boolean get_PreAuthenticate();
public void set_PreAuthenticate(
   boolean value
);
public System.Version get_ProtocolVersion();
public void set_ProtocolVersion(
   System.Version value
);
public System.Net.IWebProxy get_Proxy();
public void set_Proxy(
   System.Net.IWebProxy value
);
public System.String get_Referer();
public void set_Referer(
   System.String value
);
public System.Uri get_RequestUri();
public boolean get_SendChunked();
public void set_SendChunked(
   boolean value
);
public System.Net.ServicePoint get_ServicePoint();
public int get_Timeout();
public void set_Timeout(
   int value
);
public System.String get_TransferEncoding();
public void set_TransferEncoding(
   System.String value
);
public System.String get_UserAgent();
public void set_UserAgent(
   System.String value
);
```

Public Methods

```
public void Abort();
public void AddRange(
   int range
);
```

```
public void AddRange(
   int from,
   int to
);
public void AddRange(
   System.String rangeSpecifier,
   int range
);
public void AddRange(
   System.String rangeSpecifier,
   int from,
   int to
);
public System.IAsyncResult BeginGetRequestStream(
   System.AsyncCallback callback,
   System.Object state
);
public System.IAsyncResult BeginGetResponse(
   System.AsyncCallback callback,
   System.Object state
);
public System.IO.Stream EndGetRequestStream(
   System.IAsyncResult asyncResult
);
public System.Net.WebResponse EndGetResponse(
   System.IAsyncResult asyncResult
);
public int GetHashCode();
public System.IO.Stream GetRequestStream();
public System.Net.WebResponse GetResponse();
```

Protected Constructors

```
protected HttpWebRequest(
   System.Runtime.Serialization.SerializationInfo serializationInfo,
   System.Runtime.Serialization.StreamingContext streamingContext
);
```

HttpWebResponse Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class HttpWebResponse
   extends System.Net.WebResponse
```

Public Properties

```
public System.String get_CharacterSet();
public System.String get_ContentEncoding();
public long get_ContentLength();
public System.String get_ContentType();
public System.Net.CookieCollection get_Cookies();
public void set_Cookies(
   System.Net.CookieCollection value
);
public System.Net.WebHeaderCollection get_Headers();
public System.DateTime get_LastModified();
public System.String get_Method();
public System.Version get_ProtocolVersion();
public System.Uri get_ResponseUri();
public System.String get_Server();
public System.Net.HttpStatusCode get_StatusCode();
public System.String get_StatusDescription();
```

Public Methods

```
public void Close();
public int GetHashCode();
public System.String GetResponseHeader(
   System.String headerName
);
public System.IO.Stream GetResponseStream();
```

Protected Constructors

```
protected HttpWebResponse(
   System.Runtime.Serialization.SerializationInfo serializationInfo,
   System.Runtime.Serialization.StreamingContext streamingContext
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

IAuthenticationModule Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IAuthenticationModule
```

Public Properties

```
System.String get_AuthenticationType();
boolean get_CanPreAuthenticate();
```

Public Methods

```
System.Net.Authorization Authenticate(
   System.String challenge,
   System.Net.WebRequest request,
   System.Net.ICredentials credentials
);
System.Net.Authorization PreAuthenticate(
   System.Net.WebRequest request,
   System.Net.ICredentials credentials
);
```

ICertificatePolicy Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICertificatePolicy
```

Public Methods

```
boolean CheckValidationResult(
   System.Net.ServicePoint srvPoint,
   System.Security.Cryptography.X509Certificates.X509Certificate certificate,
   System.Net.WebRequest request,
   int certificateProblem
);
```

ICredentials Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICredentials
```

Public Methods

```
System.Net.NetworkCredential GetCredential(
   System.Uri uri,
   System.String authType
);
```

IPAddress Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class IPAddress
   extends System.Object
```

Public Constructors

```
public IPAddress(
   long newAddress
);
```

Public Properties

```
public long get_Address();
public void set_Address(
   long value
);
public System.Net.Sockets.AddressFamily get_AddressFamily();
```

Public Methods

```
public boolean Equals(
   System.Object comparand
);
public int GetHashCode();
public static short HostToNetworkOrder(
   short host
);
public static int HostToNetworkOrder(
   int host
);
public static long HostToNetworkOrder(
   long host
);
public static boolean IsLoopback(
   System.Net.IPAddress address
);
public static short NetworkToHostOrder(
   short network
);
public static int NetworkToHostOrder(
   int network
);
public static long NetworkToHostOrder(
   long network
);
public static System.Net.IPAddress Parse(
   System.String ipString
);
public System.String ToString();
```

Public Fields

```
public static final System.Net.IPAddress Any;
public static final System.Net.IPAddress Broadcast;
public static final System.Net.IPAddress Loopback;
public static final System.Net.IPAddress None;
```

IPEndPoint Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class IPEndPoint
   extends System.Net.EndPoint
```

Public Constructors

```
public IPEndPoint(
   long address,
   int port
);
public IPEndPoint(
   System.Net.IPAddress address,
   int port
);
```

Public Properties

```
public System.Net.IPAddress get_Address();
public void set_Address(
   System.Net.IPAddress value
);
public System.Net.Sockets.AddressFamily get_AddressFamily();
public int get_Port();
public void set_Port(
   int value
);
```

Public Methods

```
public System.Net.EndPoint Create(
   System.Net.SocketAddress socketAddress
);
public boolean Equals(
   System.Object comparand
);
public int GetHashCode();
public System.Net.SocketAddress Serialize();
public System.String ToString();
```

Public Fields

```
public static int MaxPort;
public static int MinPort;
```

IPHostEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class IPHostEntry
   extends System.Object
```

Public Constructors

```
public IPHostEntry();
```

Public Properties

```
public System.Net.IPAddress[] get_AddressList();
public void set_AddressList(
   System.Net.IPAddress[] value
);
public System.String[] get_Aliases();
public void set_Aliases(
   System.String[] value
);
public System.String get_HostName();
public void set_HostName(
   System.String value
);
```

IWebProxy Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IWebProxy
```

Public Properties

```
System.Net.ICredentials get_Credentials();
void set_Credentials(
   System.Net.ICredentials value
);
```

Public Methods

```
System.Uri GetProxy(
   System.Uri destination
);
boolean IsBypassed(
   System.Uri host
);
```

IWebRequestCreate Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IWebRequestCreate
```

Public Methods

```
System.Net.WebRequest Create(
   System.Uri uri
);
```

NetworkAccess Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum NetworkAccess
```

Members

Accept

Connect

NetworkCredential Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class NetworkCredential
   extends System.Object
   implements System.Net.ICredentials
```

Public Constructors

```
public NetworkCredential();
public NetworkCredential(
   System.String userName,
   System.String password
);
public NetworkCredential(
   System.String userName,
   System.String password,
   System.String domain
);
```

Public Properties

```
public System.String get_Domain();
public void set_Domain(
   System.String value
);
public System.String get_Password();
public void set_Password(
   System.String value
);
public System.String get_UserName();
public void set_UserName(
   System.String value
);
```

Public Methods

```
public final System.Net.NetworkCredential GetCredential(
   System.Uri uri,
   System.String authType
);
```

ProtocolViolationException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ProtocolViolationException
   extends System.InvalidOperationException
```

Public Constructors

```
public ProtocolViolationException();
public ProtocolViolationException(
   System.String message
);
```

Protected Constructors

```
protected ProtocolViolationException(
   System.Runtime.Serialization.SerializationInfo serializationInfo,
   System.Runtime.Serialization.StreamingContext streamingContext
);
```

ServicePoint Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ServicePoint
   extends System.Object
```

Public Properties

```
public System.Uri get_Address();
public System.Security.Cryptography.X509Certificates.X509Certificate get_Certificate();
public System.Security.Cryptography.X509Certificates.X509Certificate get_ClientCertificate();
public int get_ConnectionLimit();
public void set_ConnectionLimit(
   int value
);
public System.String get_ConnectionName();
public int get_CurrentConnections();
public System.DateTime get_IdleSince();
public int get_MaxIdleTime();
public void set_MaxIdleTime(
   int value
);
public System.Version get_ProtocolVersion();
public boolean get_SupportsPipelining();
```

Public Methods

```
public int GetHashCode();
```

ServicePointManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ServicePointManager
   extends System.Object
```

Public Properties

```
public static System.Net.ICertificatePolicy get_CertificatePolicy();
public static void set_CertificatePolicy(
   System.Net.ICertificatePolicy value
);
public static int get_DefaultConnectionLimit();
public static void set_DefaultConnectionLimit(
   int value
);
public static int get_MaxServicePointIdleTime();
public static void set_MaxServicePointIdleTime(
   int value
);
public static int get_MaxServicePoints();
public static void set_MaxServicePoints(
   int value
);
```

Public Methods

```
public static System.Net.ServicePoint FindServicePoint(
   System.Uri address
);
public static System.Net.ServicePoint FindServicePoint(
   System.String uriString,
   System.Net.IWebProxy proxy
);
public static System.Net.ServicePoint FindServicePoint(
   System.Uri address,
   System.Net.IWebProxy proxy
);
```

Public Fields

```
public static int DefaultNonPersistentConnectionLimit;
public static int DefaultPersistentConnectionLimit;
```

SocketAddress Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SocketAddress
   extends System.Object
```

Public Constructors

```
public SocketAddress(
   System.Net.Sockets.AddressFamily family
);
public SocketAddress(
   System.Net.Sockets.AddressFamily family,
   int size
);
```

Public Properties

```
public System.Net.Sockets.AddressFamily get_Family();
public ubyte get_Item(
   int offset
);
public void set_Item(
   int offset,
   ubyte value
);
public int get_Size();
```

Public Methods

```
public boolean Equals(
   System.Object comparand
);
public int GetHashCode();
public System.String ToString();
```

SocketPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SocketPermission
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Constructors

```
public SocketPermission(
   System.Security.Permissions.PermissionState state
);
public SocketPermission(
   System.Net.NetworkAccess access,
   System.Net.TransportType transport,
   System.String hostName,
   int portNumber
);
```

Public Properties

```
public System.Collections.IEnumerator get_AcceptList();
public System.Collections.IEnumerator get_ConnectList();
```

Public Methods

```
public void AddPermission(
   System.Net.NetworkAccess access,
   System.Net.TransportType transport,
   System.String hostName,
   int portNumber
);
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement securityElement
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public final boolean IsUnrestricted();
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

Public Fields

```
public static int AllPorts;
```

SocketPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class SocketPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public SocketPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.String get_Access();
public void set_Access(
   System.String value
);
public System.String get_Host();
public void set_Host(
   System.String value
);
public System.String get_Port();
public void set_Port(
   System.String value
);
public System.String get_Transport();
public void set_Transport(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

TransportType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TransportType
```

Members

All

Connectionless

ConnectionOriented

Tcp

Udp

WebClient Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class WebClient
   extends System.ComponentModel.Component
```

Public Constructors

```
public WebClient();
```

Public Properties

```
public System.String get_BaseAddress();
public void set_BaseAddress(
   System.String value
);
public System.Net.ICredentials get_Credentials();
public void set_Credentials(
   System.Net.ICredentials value
);
public System.Net.WebHeaderCollection get_Headers();
public void set_Headers(
   System.Net.WebHeaderCollection value
);
public System.Collections.Specialized.NameValueCollection get_QueryString();
public void set_QueryString(
   System.Collections.Specialized.NameValueCollection value
);
public System.Net.WebHeaderCollection get_ResponseHeaders();
```

Public Methods

```
public ubyte[] DownloadData(
   System.String address
);
public void DownloadFile(
   System.String address,
   System.String fileName
);
public System.IO.Stream OpenRead(
   System.String address
);
public System.IO.Stream OpenWrite(
   System.String address
);
public System.IO.Stream OpenWrite(
   System.String address,
   System.String method
);
public ubyte[] UploadData(
   System.String address,
   ubyte[] data
);
public ubyte[] UploadData(
   System.String address,
```

```
      System.String method,
      ubyte[] data
   );
   public ubyte[] UploadFile(
      System.String address,
      System.String fileName
   );
   public ubyte[] UploadFile(
      System.String address,
      System.String method,
      System.String fileName
   );
   public ubyte[] UploadValues(
      System.String address,
      System.Collections.Specialized.NameValueCollection data
   );
   public ubyte[] UploadValues(
      System.String address,
      System.String method,
      System.Collections.Specialized.NameValueCollection data
   );
```

WebException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class WebException
   extends System.InvalidOperationException
```

Public Constructors

```
public WebException();
public WebException(
   System.String message
);
public WebException(
   System.String message,
   System.Exception innerException
);
public WebException(
   System.String message,
   System.Net.WebExceptionStatus status
);
public WebException(
   System.String message,
   System.Exception innerException,
   System.Net.WebExceptionStatus status,
   System.Net.WebResponse response
);
```

Public Properties

```
public System.Net.WebResponse get_Response();
public System.Net.WebExceptionStatus get_Status();
```

Protected Constructors

```
protected WebException(
   System.Runtime.Serialization.SerializationInfo serializationInfo,
   System.Runtime.Serialization.StreamingContext streamingContext
);
```

WebExceptionStatus Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum WebExceptionStatus
```

Members

ConnectFailure

ConnectionClosed

KeepAliveFailure

NameResolutionFailure

Pending

PipelineFailure

ProtocolError

ProxyNameResolutionFailure

ReceiveFailure

RequestCanceled

SecureChannelFailure

SendFailure

ServerProtocolViolation

Success

Timeout

TrustFailure

WebHeaderCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class WebHeaderCollection
   extends System.Collections.Specialized.NameValueCollection
```

Public Constructors

```
public WebHeaderCollection();
```

Public Methods

```
public void Add(
   System.String header
);
public void Add(
   System.String name,
   System.String value
);
public System.String[] GetValues(
   System.String header
);
public static boolean IsRestricted(
   System.String headerName
);
public void OnDeserialization(
   System.Object sender
);
public void Remove(
   System.String name
);
public void Set(
   System.String name,
   System.String value
);
public ubyte[] ToByteArray();
public System.String ToString();
```

Protected Constructors

```
protected WebHeaderCollection(
   System.Runtime.Serialization.SerializationInfo serializationInfo,
   System.Runtime.Serialization.StreamingContext streamingContext
);
```

Protected Methods

```
protected void AddWithoutValidate(
   System.String headerName,
   System.String headerValue
);
```

WebPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class WebPermission
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Constructors

```
public WebPermission();
public WebPermission(
   System.Security.Permissions.PermissionState state
);
public WebPermission(
   System.Net.NetworkAccess access,
   System.String uriString
);
public WebPermission(
   System.Net.NetworkAccess access,
   System.Text.RegularExpressions.Regex uriRegex
);
```

Public Properties

```
public System.Collections.IEnumerator get_AcceptList();
public System.Collections.IEnumerator get_ConnectList();
```

Public Methods

```
public void AddPermission(
   System.Net.NetworkAccess access,
   System.String uriString
);
public void AddPermission(
   System.Net.NetworkAccess access,
   System.Text.RegularExpressions.Regex uriRegex
);
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement securityElement
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public final boolean IsUnrestricted();
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

WebPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class WebPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public WebPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.String get_Accept();
public void set_Accept(
   System.String value
);
public System.String get_AcceptPattern();
public void set_AcceptPattern(
   System.String value
);
public System.String get_Connect();
public void set_Connect(
   System.String value
);
public System.String get_ConnectPattern();
public void set_ConnectPattern(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

WebProxy Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class WebProxy
   extends System.Object
   implements System.Net.IWebProxy, System.Runtime.Serialization.ISerializable
```

Public Constructors

```
public WebProxy();
public WebProxy(
   System.String Address
);
public WebProxy(
   System.Uri Address
);
public WebProxy(
   System.String Address,
   boolean BypassOnLocal
);
public WebProxy(
   System.String Host,
   int Port
);
public WebProxy(
   System.Uri Address,
   boolean BypassOnLocal
);
public WebProxy(
   System.String Address,
   boolean BypassOnLocal,
   System.String[] BypassList
);
public WebProxy(
   System.Uri Address,
   boolean BypassOnLocal,
   System.String[] BypassList
);
public WebProxy(
   System.String Address,
   boolean BypassOnLocal,
   System.String[] BypassList,
   System.Net.ICredentials Credentials
);
public WebProxy(
   System.Uri Address,
   boolean BypassOnLocal,
   System.String[] BypassList,
   System.Net.ICredentials Credentials
);
```

Public Properties

```
public System.Uri get_Address();
public void set_Address(
   System.Uri value
);
```

```
public System.Collections.ArrayList get_BypassArrayList();
public System.String[] get_BypassList();
public void set_BypassList(
   System.String[] value
);
public boolean get_BypassProxyOnLocal();
public void set_BypassProxyOnLocal(
   boolean value
);
public final System.Net.ICredentials get_Credentials();
public final void set_Credentials(
   System.Net.ICredentials value
);
```

Public Methods

```
public static System.Net.WebProxy GetDefaultProxy();
public final System.Uri GetProxy(
   System.Uri destination
);
public final boolean IsBypassed(
   System.Uri host
);
```

Protected Constructors

```
protected WebProxy(
   System.Runtime.Serialization.SerializationInfo serializationInfo,
   System.Runtime.Serialization.StreamingContext streamingContext
);
```

WebRequest Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class WebRequest
   extends System.MarshalByRefObject
   implements System.Runtime.Serialization.ISerializable
```

Public Properties

```
public System.String get_ConnectionGroupName();
public void set_ConnectionGroupName(
   System.String value
);
public long get_ContentLength();
public void set_ContentLength(
   long value
);
public System.String get_ContentType();
public void set_ContentType(
   System.String value
);
public System.Net.ICredentials get_Credentials();
public void set_Credentials(
   System.Net.ICredentials value
);
public System.Net.WebHeaderCollection get_Headers();
public void set_Headers(
   System.Net.WebHeaderCollection value
);
public System.String get_Method();
public void set_Method(
   System.String value
);
public boolean get_PreAuthenticate();
public void set_PreAuthenticate(
   boolean value
);
public System.Net.IWebProxy get_Proxy();
public void set_Proxy(
   System.Net.IWebProxy value
);
public System.Uri get_RequestUri();
public int get_Timeout();
public void set_Timeout(
   int value
);
```

Public Methods

```
public void Abort();
public System.IAsyncResult BeginGetRequestStream(
   System.AsyncCallback callback,
   System.Object state
);
public System.IAsyncResult BeginGetResponse(
   System.AsyncCallback callback,
   System.Object state
```

```
);
public static System.Net.WebRequest Create(
   System.String requestUriString
);
public static System.Net.WebRequest Create(
   System.Uri requestUri
);
public static System.Net.WebRequest CreateDefault(
   System.Uri requestUri
);
public System.IO.Stream EndGetRequestStream(
   System.IAsyncResult asyncResult
);
public System.Net.WebResponse EndGetResponse(
   System.IAsyncResult asyncResult
);
public System.IO.Stream GetRequestStream();
public System.Net.WebResponse GetResponse();
public static boolean RegisterPrefix(
   System.String prefix,
   System.Net.IWebRequestCreate creator
);
```

Protected Constructors

```
protected WebRequest();
protected WebRequest(
   System.Runtime.Serialization.SerializationInfo serializationInfo,
   System.Runtime.Serialization.StreamingContext streamingContext
);
```

WebResponse Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class WebResponse
   extends System.MarshalByRefObject
   implements System.Runtime.Serialization.ISerializable, System.IDisposable
```

Public Properties

```
public long get_ContentLength();
public void set_ContentLength(
   long value
);
public System.String get_ContentType();
public void set_ContentType(
   System.String value
);
public System.Net.WebHeaderCollection get_Headers();
public System.Uri get_ResponseUri();
```

Public Methods

```
public void Close();
public System.IO.Stream GetResponseStream();
```

Protected Constructors

```
protected WebResponse();
protected WebResponse(
   System.Runtime.Serialization.SerializationInfo serializationInfo,
   System.Runtime.Serialization.StreamingContext streamingContext
);
```

System.Net.Sockets Namespace (Visual J# Syntax)

Classes

LingerOption

MulticastOption

NetworkStream

Socket

SocketException

TcpClient

TcpListener

UdpClient

Enumerations

AddressFamily

ProtocolFamily

ProtocolType

SelectMode

SocketFlags

SocketOptionLevel

SocketOptionName

SocketShutdown

SocketType

AddressFamily Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum AddressFamily
```

Members

AppleTalk

Atm

Banyan

Ccitt

Chaos

Cluster

DataKit

DataLink

DecNet

Ecma

FireFox

HyperChannel

Ieee12844

ImpLink

InterNetwork

InterNetworkV6

Ipx

Irda

Iso

Lat

Max

NetBios

NetworkDesigners

NS

Osi

Pup

Sna

Unix

Unknown

Unspecified

LingerOption Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LingerOption
   extends System.Object
```

Public Constructors

```
public LingerOption(
   boolean enable,
   int seconds
);
```

Public Properties

```
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public int get_LingerTime();
public void set_LingerTime(
   int value
);
```

MulticastOption Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MulticastOption
   extends System.Object
```

Public Constructors

```
public MulticastOption(
   System.Net.IPAddress group
);
public MulticastOption(
   System.Net.IPAddress group,
   System.Net.IPAddress mcint
);
```

Public Properties

```
public System.Net.IPAddress get_Group();
public void set_Group(
   System.Net.IPAddress value
);
public System.Net.IPAddress get_LocalAddress();
public void set_LocalAddress(
   System.Net.IPAddress value
);
```

NetworkStream Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class NetworkStream
   extends System.IO.Stream
```

Public Constructors

```
public NetworkStream(
   System.Net.Sockets.Socket socket
);
public NetworkStream(
   System.Net.Sockets.Socket socket,
   boolean ownsSocket
);
public NetworkStream(
   System.Net.Sockets.Socket socket,
   System.IO.FileAccess access
);
public NetworkStream(
   System.Net.Sockets.Socket socket,
   System.IO.FileAccess access,
   boolean ownsSocket
);
```

Public Properties

```
public boolean get_CanRead();
public boolean get_CanSeek();
public boolean get_CanWrite();
public boolean get_DataAvailable();
public long get_Length();
public long get_Position();
public void set_Position(
   long value
);
```

Public Methods

```
public System.IAsyncResult BeginRead(
   ubyte[] buffer,
   int offset,
   int size,
   System.AsyncCallback callback,
   System.Object state
);
public System.IAsyncResult BeginWrite(
   ubyte[] buffer,
   int offset,
   int size,
   System.AsyncCallback callback,
   System.Object state
);
public void Close();
public int EndRead(
   System.IAsyncResult asyncResult
```

```
);
public void EndWrite(
   System.IAsyncResult asyncResult
);
public void Flush();
public int Read(
   ubyte[] buffer,
   int offset,
   int size
);
public long Seek(
   long offset,
   System.IO.SeekOrigin origin
);
public void SetLength(
   long value
);
public void Write(
   ubyte[] buffer,
   int offset,
   int size
);
```

Protected Properties

```
protected boolean get_Readable();
protected void set_Readable(
   boolean value
);
protected System.Net.Sockets.Socket get_Socket();
protected boolean get_Writeable();
protected void set_Writeable(
   boolean value
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

ProtocolFamily Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ProtocolFamily
```

Members

AppleTalk

Atm

Banyan

Ccitt

Chaos

Cluster

DataKit

DataLink

DecNet

Ecma

FireFox

HyperChannel

Ieee12844

ImpLink

InterNetwork

InterNetworkV6

Ipx

Irda

Iso

Lat

Max

NetBios

NetworkDesigners

NS

Osi

Pup

Sna

Unix

Unknown

Unspecified

ProtocolType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ProtocolType
```

Members

Ggp

Icmp

Idp

Igmp

IP

Ipx

ND

Pup

Raw

Spx

SpxII

Tcp

Udp

Unknown

Unspecified

SelectMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SelectMode
```

Members

SelectError

SelectRead

SelectWrite

Socket Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Socket
   extends System.Object
   implements System.IDisposable
```

Public Constructors

```
public Socket(
   System.Net.Sockets.AddressFamily addressFamily,
   System.Net.Sockets.SocketType socketType,
   System.Net.Sockets.ProtocolType protocolType
);
```

Public Properties

```
public System.Net.Sockets.AddressFamily get_AddressFamily();
public int get_Available();
public boolean get_Blocking();
public void set_Blocking(
   boolean value
);
public boolean get_Connected();
public System.IntPtr get_Handle();
public System.Net.EndPoint get_LocalEndPoint();
public System.Net.Sockets.ProtocolType get_ProtocolType();
public System.Net.EndPoint get_RemoteEndPoint();
public System.Net.Sockets.SocketType get_SocketType();
```

Public Methods

```
public System.Net.Sockets.Socket Accept();
public System.IAsyncResult BeginAccept(
   System.AsyncCallback callback,
   System.Object state
);
public System.IAsyncResult BeginConnect(
   System.Net.EndPoint remoteEP,
   System.AsyncCallback callback,
   System.Object state
);
public System.IAsyncResult BeginReceive(
   ubyte[] buffer,
   int offset,
   int size,
   System.Net.Sockets.SocketFlags socketFlags,
   System.AsyncCallback callback,
   System.Object state
);
public System.IAsyncResult BeginReceiveFrom(
   ubyte[] buffer,
   int offset,
   int size,
   System.Net.Sockets.SocketFlags socketFlags,
   System.Net.EndPoint& remoteEP,
```

```
   System.AsyncCallback callback,
   System.Object state
);
public System.IAsyncResult BeginSend(
   ubyte[] buffer,
   int offset,
   int size,
   System.Net.Sockets.SocketFlags socketFlags,
   System.AsyncCallback callback,
   System.Object state
);
public System.IAsyncResult BeginSendTo(
   ubyte[] buffer,
   int offset,
   int size,
   System.Net.Sockets.SocketFlags socketFlags,
   System.Net.EndPoint remoteEP,
   System.AsyncCallback callback,
   System.Object state
);
public void Bind(
   System.Net.EndPoint localEP
);
public void Close();
public void Connect(
   System.Net.EndPoint remoteEP
);
public System.Net.Sockets.Socket EndAccept(
   System.IAsyncResult asyncResult
);
public void EndConnect(
   System.IAsyncResult asyncResult
);
public int EndReceive(
   System.IAsyncResult asyncResult
);
public int EndReceiveFrom(
   System.IAsyncResult asyncResult,
   System.Net.EndPoint& endPoint
);
public int EndSend(
   System.IAsyncResult asyncResult
);
public int EndSendTo(
   System.IAsyncResult asyncResult
);
public int GetHashCode();
public System.Object GetSocketOption(
   System.Net.Sockets.SocketOptionLevel optionLevel,
   System.Net.Sockets.SocketOptionName optionName
);
public ubyte[] GetSocketOption(
   System.Net.Sockets.SocketOptionLevel optionLevel,
   System.Net.Sockets.SocketOptionName optionName,
   int optionLength
);
public void GetSocketOption(
   System.Net.Sockets.SocketOptionLevel optionLevel,
   System.Net.Sockets.SocketOptionName optionName,
   ubyte[] optionValue
);
public int IOControl(
   int ioControlCode,
   ubyte[] optionInValue,
   ubyte[] optionOutValue
);
public void Listen(
   int backlog
```

```
);
public boolean Poll(
   int microSeconds,
   System.Net.Sockets.SelectMode mode
);
public int Receive(
   ubyte[] buffer
);
public int Receive(
   ubyte[] buffer,
   System.Net.Sockets.SocketFlags socketFlags
);
public int Receive(
   ubyte[] buffer,
   int size,
   System.Net.Sockets.SocketFlags socketFlags
);
public int Receive(
   ubyte[] buffer,
   int offset,
   int size,
   System.Net.Sockets.SocketFlags socketFlags
);
public int ReceiveFrom(
   ubyte[] buffer,
   System.Net.EndPoint& remoteEP
);
public int ReceiveFrom(
   ubyte[] buffer,
   System.Net.Sockets.SocketFlags socketFlags,
   System.Net.EndPoint& remoteEP
);
public int ReceiveFrom(
   ubyte[] buffer,
   int size,
   System.Net.Sockets.SocketFlags socketFlags,
   System.Net.EndPoint& remoteEP
);
public int ReceiveFrom(
   ubyte[] buffer,
   int offset,
   int size,
   System.Net.Sockets.SocketFlags socketFlags,
   System.Net.EndPoint& remoteEP
);
public static void Select(
   System.Collections.IList checkRead,
   System.Collections.IList checkWrite,
   System.Collections.IList checkError,
   int microSeconds
);
public int Send(
   ubyte[] buffer
);
public int Send(
   ubyte[] buffer,
   System.Net.Sockets.SocketFlags socketFlags
);
public int Send(
   ubyte[] buffer,
   int size,
   System.Net.Sockets.SocketFlags socketFlags
);
public int Send(
   ubyte[] buffer,
   int offset,
   int size,
   System.Net.Sockets.SocketFlags socketFlags
```

```
);
public int SendTo(
   ubyte[] buffer,
   System.Net.EndPoint remoteEP
);
public int SendTo(
   ubyte[] buffer,
   System.Net.Sockets.SocketFlags socketFlags,
   System.Net.EndPoint remoteEP
);
public int SendTo(
   ubyte[] buffer,
   int size,
   System.Net.Sockets.SocketFlags socketFlags,
   System.Net.EndPoint remoteEP
);
public int SendTo(
   ubyte[] buffer,
   int offset,
   int size,
   System.Net.Sockets.SocketFlags socketFlags,
   System.Net.EndPoint remoteEP
);
public void SetSocketOption(
   System.Net.Sockets.SocketOptionLevel optionLevel,
   System.Net.Sockets.SocketOptionName optionName,
   ubyte[] optionValue
);
public void SetSocketOption(
   System.Net.Sockets.SocketOptionLevel optionLevel,
   System.Net.Sockets.SocketOptionName optionName,
   int optionValue
);
public void SetSocketOption(
   System.Net.Sockets.SocketOptionLevel optionLevel,
   System.Net.Sockets.SocketOptionName optionName,
   System.Object optionValue
);
public void Shutdown(
   System.Net.Sockets.SocketShutdown how
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

SocketException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SocketException
   extends System.ComponentModel.Win32Exception
```

Public Constructors

```
public SocketException();
public SocketException(
   int errorCode
);
```

Public Properties

```
public int get_ErrorCode();
```

Protected Constructors

```
protected SocketException(
   System.Runtime.Serialization.SerializationInfo serializationInfo,
   System.Runtime.Serialization.StreamingContext streamingContext
);
```

SocketFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum SocketFlags
```

Members

DontRoute

MaxIOVectorLength

None

OutOfBand

Partial

Peek

SocketOptionLevel Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SocketOptionLevel
```

Members

IP

Socket

Tcp

Udp

SocketOptionName Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SocketOptionName
```

Members

AcceptConnection

AddMembership

AddSourceMembership

BlockSource

Broadcast

BsdUrgent

ChecksumCoverage

Debug

DontFragment

DontLinger

DontRoute

DropMembership

DropSourceMembership

Error

ExclusiveAddressUse

Expedited

HeaderIncluded

IPOptions

IpTimeToLive

KeepAlive

Linger

MaxConnections

MulticastInterface

MulticastLoopback

MulticastTimeToLive

NoChecksum

NoDelay

OutOfBandInline

PacketInformation

ReceiveBuffer

ReceiveLowWater

ReceiveTimeout

ReuseAddress

SendBuffer

SendLowWater

SendTimeout

Type

TypeOfService

UnblockSource

UseLoopback

SocketShutdown Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SocketShutdown
```

Members

Both

Receive

Send

SocketType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SocketType
```

Members

Dgram

Raw

Rdm

Seqpacket

Stream

Unknown

TcpClient Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TcpClient
   extends System.Object
   implements System.IDisposable
```

Public Constructors

```
public TcpClient();
public TcpClient(
   System.Net.IPEndPoint localEP
);
public TcpClient(
   System.String hostname,
   int port
);
```

Public Properties

```
public System.Net.Sockets.LingerOption get_LingerState();
public void set_LingerState(
   System.Net.Sockets.LingerOption value
);
public boolean get_NoDelay();
public void set_NoDelay(
   boolean value
);
public int get_ReceiveBufferSize();
public void set_ReceiveBufferSize(
   int value
);
public int get_ReceiveTimeout();
public void set_ReceiveTimeout(
   int value
);
public int get_SendBufferSize();
public void set_SendBufferSize(
   int value
);
public int get_SendTimeout();
public void set_SendTimeout(
   int value
);
```

Public Methods

```
public void Close();
public void Connect(
   System.Net.IPEndPoint remoteEP
);
public void Connect(
   System.Net.IPAddress address,
   int port
);
public void Connect(
```

```
   System.String hostname,
   int port
);
public System.Net.Sockets.NetworkStream GetStream();
```

Protected Properties

```
protected boolean get_Active();
protected void set_Active(
   boolean value
);
protected System.Net.Sockets.Socket get_Client();
protected void set_Client(
   System.Net.Sockets.Socket value
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

TcpListener Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TcpListener
   extends System.Object
```

Public Constructors

```
public TcpListener(
   int port
);
public TcpListener(
   System.Net.IPEndPoint localEP
);
public TcpListener(
   System.Net.IPAddress localaddr,
   int port
);
```

Public Properties

```
public System.Net.EndPoint get_LocalEndpoint();
```

Public Methods

```
public System.Net.Sockets.Socket AcceptSocket();
public System.Net.Sockets.TcpClient AcceptTcpClient();
public boolean Pending();
public void Start();
public void Stop();
```

Protected Properties

```
protected boolean get_Active();
protected System.Net.Sockets.Socket get_Server();
```

Protected Methods

```
protected void Finalize();
```

UdpClient Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UdpClient
   extends System.Object
   implements System.IDisposable
```

Public Constructors

```
public UdpClient();
public UdpClient(
   int port
);
public UdpClient(
   System.Net.IPEndPoint localEP
);
public UdpClient(
   System.String hostname,
   int port
);
```

Public Methods

```
public void Close();
public void Connect(
   System.Net.IPEndPoint endPoint
);
public void Connect(
   System.Net.IPAddress addr,
   int port
);
public void Connect(
   System.String hostname,
   int port
);
public void DropMulticastGroup(
   System.Net.IPAddress multicastAddr
);
public void JoinMulticastGroup(
   System.Net.IPAddress multicastAddr
);
public void JoinMulticastGroup(
   System.Net.IPAddress multicastAddr,
   int timeToLive
);
public ubyte[] Receive(
   System.Net.IPEndPoint& remoteEP
);
public int Send(
   ubyte[] dgram,
   int bytes
);
public int Send(
   ubyte[] dgram,
   int bytes,
   System.Net.IPEndPoint endPoint
);
public int Send(
```

```
   ubyte[] dgram,
   int bytes,
   System.String hostname,
   int port
);
```

Protected Properties

```
protected boolean get_Active();
protected void set_Active(
   boolean value
);
protected System.Net.Sockets.Socket get_Client();
protected void set_Client(
   System.Net.Sockets.Socket value
);
```

System.Reflection Namespace (Visual J# Syntax)

Classes

AmbiguousMatchException

Assembly

AssemblyAlgorithmIdAttribute

AssemblyCompanyAttribute

AssemblyConfigurationAttribute

AssemblyCopyrightAttribute

AssemblyCultureAttribute

AssemblyDefaultAliasAttribute

AssemblyDelaySignAttribute

AssemblyDescriptionAttribute

AssemblyFileVersionAttribute

AssemblyFlagsAttribute

AssemblyInformationalVersionAttribute

AssemblyKeyFileAttribute

AssemblyKeyNameAttribute

AssemblyName

AssemblyNameProxy

AssemblyProductAttribute

AssemblyTitleAttribute

AssemblyTrademarkAttribute

AssemblyVersionAttribute

Binder

ConstructorInfo

CustomAttributeFormatException

DefaultMemberAttribute

EventInfo

FieldInfo

InvalidFilterCriteriaException

ManifestResourceInfo

MemberInfo

MethodBase

MethodInfo

Missing

Module

ParameterInfo

Pointer

PropertyInfo

ReflectionTypeLoadException

StrongNameKeyPair

TargetException

TargetInvocationException

TargetParameterCountException

TypeDelegator

Interfaces

ICustomAttributeProvider

IReflect

Structures

InterfaceMapping

ParameterModifier

Delegates

MemberFilter

ModuleResolveEventHandler

TypeFilter

Enumerations

AssemblyNameFlags

BindingFlags

CallingConventions

EventAttributes

FieldAttributes

MemberTypes

MethodAttributes

MethodImplAttributes

ParameterAttributes

PropertyAttributes

ResourceAttributes

ResourceLocation

TypeAttributes

AmbiguousMatchException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class AmbiguousMatchException
   extends System.SystemException
```

Public Constructors

```
public AmbiguousMatchException();
public AmbiguousMatchException(
   System.String message
);
public AmbiguousMatchException(
   System.String message,
   System.Exception inner
);
```

Assembly Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDual) */
public class Assembly
   extends System.Object
   implements System.Security.IEvidenceFactory, System.Reflection.ICustomAttributeProvider, S
ystem.Runtime.Serialization.ISerializable
```

Public Properties

```
public System.String get_CodeBase();
public System.Reflection.MethodInfo get_EntryPoint();
public System.String get_EscapedCodeBase();
public System.Security.Policy.Evidence get_Evidence();
public System.String get_FullName();
public boolean get_GlobalAssemblyCache();
public System.String get_Location();
```

Public Methods

```
public System.Object CreateInstance(
   System.String typeName
);
public System.Object CreateInstance(
   System.String typeName,
   boolean ignoreCase
);
public System.Object CreateInstance(
   System.String typeName,
   boolean ignoreCase,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Object[] args,
   System.Globalization.CultureInfo culture,
   System.Object[] activationAttributes
);
public static System.String CreateQualifiedName(
   System.String assemblyName,
   System.String typeName
);
public static System.Reflection.Assembly GetAssembly(
   System.Type type
);
public static System.Reflection.Assembly GetCallingAssembly();
public System.Object[] GetCustomAttributes(
   boolean inherit
);
public System.Object[] GetCustomAttributes(
   System.Type attributeType,
   boolean inherit
);
public static System.Reflection.Assembly GetEntryAssembly();
public static System.Reflection.Assembly GetExecutingAssembly();
public System.Type[] GetExportedTypes();
public System.IO.FileStream GetFile(
   System.String name
```

```
);
public System.IO.FileStream[] GetFiles();
public System.IO.FileStream[] GetFiles(
   boolean getResourceModules
);
public System.Reflection.Module[] GetLoadedModules();
public System.Reflection.Module[] GetLoadedModules(
   boolean getResourceModules
);
public System.Reflection.ManifestResourceInfo GetManifestResourceInfo(
   System.String resourceName
);
public System.String[] GetManifestResourceNames();
public System.IO.Stream GetManifestResourceStream(
   System.String name
);
public System.IO.Stream GetManifestResourceStream(
   System.Type type,
   System.String name
);
public System.Reflection.Module GetModule(
   System.String name
);
public System.Reflection.Module[] GetModules();
public System.Reflection.Module[] GetModules(
   boolean getResourceModules
);
public System.Reflection.AssemblyName GetName();
public System.Reflection.AssemblyName GetName(
   boolean copiedName
);
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public System.Reflection.AssemblyName[] GetReferencedAssemblies();
public System.Reflection.Assembly GetSatelliteAssembly(
   System.Globalization.CultureInfo culture
);
public System.Reflection.Assembly GetSatelliteAssembly(
   System.Globalization.CultureInfo culture,
   System.Version version
);
public System.Type GetType(
   System.String name
);
public System.Type GetType(
   System.String name,
   boolean throwOnError
);
public System.Type GetType(
   System.String name,
   boolean throwOnError,
   boolean ignoreCase
);
public System.Type[] GetTypes();
public boolean IsDefined(
   System.Type attributeType,
   boolean inherit
);
public static System.Reflection.Assembly Load(
   ubyte[] rawAssembly
);
public static System.Reflection.Assembly Load(
   System.Reflection.AssemblyName assemblyRef
);
public static System.Reflection.Assembly Load(
   System.String assemblyString
```

```
);
public static System.Reflection.Assembly Load(
   ubyte[] rawAssembly,
   ubyte[] rawSymbolStore
);
public static System.Reflection.Assembly Load(
   System.Reflection.AssemblyName assemblyRef,
   System.Security.Policy.Evidence assemblySecurity
);
public static System.Reflection.Assembly Load(
   System.String assemblyString,
   System.Security.Policy.Evidence assemblySecurity
);
public static System.Reflection.Assembly Load(
   ubyte[] rawAssembly,
   ubyte[] rawSymbolStore,
   System.Security.Policy.Evidence securityEvidence
);
public static System.Reflection.Assembly LoadFrom(
   System.String assemblyFile
);
public static System.Reflection.Assembly LoadFrom(
   System.String assemblyFile,
   System.Security.Policy.Evidence securityEvidence
);
public System.Reflection.Module LoadModule(
   System.String moduleName,
   ubyte[] rawModule
);
public System.Reflection.Module LoadModule(
   System.String moduleName,
   ubyte[] rawModule,
   ubyte[] rawSymbolStore
);
public static System.Reflection.Assembly LoadWithPartialName(
   System.String partialName
);
public static System.Reflection.Assembly LoadWithPartialName(
   System.String partialName,
   System.Security.Policy.Evidence securityEvidence
);
public System.String ToString();
```

Public Events

```
public void add_ModuleResolve(
   System.Reflection.ModuleResolveEventHandler value
);
public void remove_ModuleResolve(
   System.Reflection.ModuleResolveEventHandler value
);
```

AssemblyAlgorithmIdAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyAlgorithmIdAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyAlgorithmIdAttribute(
   System.Configuration.Assemblies.AssemblyHashAlgorithm algorithmId
);
```

Public Properties

```
public System.UInt32 get_AlgorithmId();
```

AssemblyCompanyAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyCompanyAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyCompanyAttribute(
   System.String company
);
```

Public Properties

```
public System.String get_Company();
```

AssemblyConfigurationAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyConfigurationAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyConfigurationAttribute(
   System.String configuration
);
```

Public Properties

```
public System.String get_Configuration();
```

AssemblyCopyrightAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyCopyrightAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyCopyrightAttribute(
   System.String copyright
);
```

Public Properties

```
public System.String get_Copyright();
```

AssemblyCultureAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyCultureAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyCultureAttribute(
   System.String culture
);
```

Public Properties

```
public System.String get_Culture();
```

AssemblyDefaultAliasAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyDefaultAliasAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyDefaultAliasAttribute(
   System.String defaultAlias
);
```

Public Properties

```
public System.String get_DefaultAlias();
```

AssemblyDelaySignAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyDelaySignAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyDelaySignAttribute(
   boolean delaySign
);
```

Public Properties

```
public boolean get_DelaySign();
```

AssemblyDescriptionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyDescriptionAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyDescriptionAttribute(
   System.String description
);
```

Public Properties

```
public System.String get_Description();
```

AssemblyFileVersionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyFileVersionAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyFileVersionAttribute(
   System.String version
);
```

Public Properties

```
public System.String get_Version();
```

AssemblyFlagsAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyFlagsAttribute
   extends System.Attribute
```

Public Properties

```
public System.UInt32 get_Flags();
```

AssemblyInformationalVersionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyInformationalVersionAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyInformationalVersionAttribute(
   System.String informationalVersion
);
```

Public Properties

```
public System.String get_InformationalVersion();
```

AssemblyKeyFileAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyKeyFileAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyKeyFileAttribute(
   System.String keyFile
);
```

Public Properties

```
public System.String get_KeyFile();
```

AssemblyKeyNameAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyKeyNameAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyKeyNameAttribute(
   System.String keyName
);
```

Public Properties

```
public System.String get_KeyName();
```

AssemblyName Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class AssemblyName
   extends System.Object
   implements System.ICloneable, System.Runtime.Serialization.ISerializable, System.Runtime.S
erialization.IDeserializationCallback
```

Public Constructors

```
public AssemblyName();
```

Public Properties

```
public System.String get_CodeBase();
public void set_CodeBase(
   System.String value
);
public System.Globalization.CultureInfo get_CultureInfo();
public void set_CultureInfo(
   System.Globalization.CultureInfo value
);
public System.String get_EscapedCodeBase();
public System.Reflection.AssemblyNameFlags get_Flags();
public void set_Flags(
   System.Reflection.AssemblyNameFlags value
);
public System.String get_FullName();
public System.Configuration.Assemblies.AssemblyHashAlgorithm get_HashAlgorithm();
public void set_HashAlgorithm(
   System.Configuration.Assemblies.AssemblyHashAlgorithm value
);
public System.Reflection.StrongNameKeyPair get_KeyPair();
public void set_KeyPair(
   System.Reflection.StrongNameKeyPair value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Version get_Version();
public void set_Version(
   System.Version value
);
public System.Configuration.Assemblies.AssemblyVersionCompatibility get_VersionCompatibility(
);
public void set_VersionCompatibility(
   System.Configuration.Assemblies.AssemblyVersionCompatibility value
);
```

Public Methods

```
public final System.Object Clone();
public static System.Reflection.AssemblyName GetAssemblyName(
   System.String assemblyFile
```

```
);
public final void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public ubyte[] GetPublicKey();
public ubyte[] GetPublicKeyToken();
public final void OnDeserialization(
   System.Object sender
);
public void SetPublicKey(
   ubyte[] publicKey
);
public void SetPublicKeyToken(
   ubyte[] publicKeyToken
);
public System.String ToString();
```

AssemblyNameFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum AssemblyNameFlags
```

Members

None

PublicKey

AssemblyNameProxy Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AssemblyNameProxy
   extends System.MarshalByRefObject
```

Public Constructors

```
public AssemblyNameProxy();
```

Public Methods

```
public System.Reflection.AssemblyName GetAssemblyName(
   System.String assemblyFile
);
```

AssemblyProductAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyProductAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyProductAttribute(
   System.String product
);
```

Public Properties

```
public System.String get_Product();
```

AssemblyTitleAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyTitleAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyTitleAttribute(
   System.String title
);
```

Public Properties

```
public System.String get_Title();
```

AssemblyTrademarkAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyTrademarkAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyTrademarkAttribute(
   System.String trademark
);
```

Public Properties

```
public System.String get_Trademark();
```

AssemblyVersionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class AssemblyVersionAttribute
   extends System.Attribute
```

Public Constructors

```
public AssemblyVersionAttribute(
   System.String version
);
```

Public Properties

```
public System.String get_Version();
```

Binder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDual) */
public abstract class Binder
   extends System.Object
```

Public Methods

```
public abstract System.Reflection.FieldInfo BindToField(
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.FieldInfo[] match,
   System.Object value,
   System.Globalization.CultureInfo culture
);
public abstract System.Reflection.MethodBase BindToMethod(
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.MethodBase[] match,
   System.Object[]& args,
   System.Reflection.ParameterModifier[] modifiers,
   System.Globalization.CultureInfo culture,
   System.String[] names,
   System.Object& state
);
public abstract System.Object ChangeType(
   System.Object value,
   System.Type type,
   System.Globalization.CultureInfo culture
);
public abstract void ReorderArgumentArray(
   System.Object[]& args,
   System.Object state
);
public abstract System.Reflection.MethodBase SelectMethod(
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.MethodBase[] match,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
public abstract System.Reflection.PropertyInfo SelectProperty(
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.PropertyInfo[] match,
   System.Type returnType,
   System.Type[] indexes,
   System.Reflection.ParameterModifier[] modifiers
);
```

Protected Constructors

```
protected Binder();
```

BindingFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum BindingFlags
```

Members

CreateInstance

DeclaredOnly

Default

ExactBinding

FlattenHierarchy

GetField

GetProperty

IgnoreCase

IgnoreReturn

Instance

InvokeMethod

NonPublic

OptionalParamBinding

Public

PutDispProperty

PutRefDispProperty

SetField

SetProperty

Static

SuppressChangeType

CallingConventions Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum CallingConventions
```

Members

Any

ExplicitThis

HasThis

Standard

VarArgs

ConstructorInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDual) */
public abstract class ConstructorInfo
   extends System.Reflection.MethodBase
```

Public Properties

```
public System.Reflection.MemberTypes get_MemberType();
```

Public Methods

```
/** @attribute System.Diagnostics.DebuggerStepThroughAttribute() */
/** @attribute System.Diagnostics.DebuggerHiddenAttribute() */
public System.Object Invoke(
   System.Object[] parameters
);
public abstract System.Object Invoke(
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Object[] parameters,
   System.Globalization.CultureInfo culture
);
```

Public Fields

```
public static final System.String ConstructorName;
public static final System.String TypeConstructorName;
```

Protected Constructors

```
protected ConstructorInfo();
```

CustomAttributeFormatException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class CustomAttributeFormatException
   extends System.FormatException
```

Public Constructors

```
public CustomAttributeFormatException();
public CustomAttributeFormatException(
   System.String message
);
public CustomAttributeFormatException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected CustomAttributeFormatException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

DefaultMemberAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, Interface, AllowMultip
le = false, Inherited = true) */
public class DefaultMemberAttribute
   extends System.Attribute
```

Public Constructors

```
public DefaultMemberAttribute(
   System.String memberName
);
```

Public Properties

```
public System.String get_MemberName();
```

EventAttributes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum EventAttributes
```

Members

None

ReservedMask

RTSpecialName

SpecialName

EventInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDual) */
public abstract class EventInfo
   extends System.Reflection.MemberInfo
```

Public Properties

```
public abstract System.Reflection.EventAttributes get_Attributes();
public System.Type get_EventHandlerType();
public boolean get_IsMulticast();
public boolean get_IsSpecialName();
public System.Reflection.MemberTypes get_MemberType();
```

Public Methods

```
/** @attribute System.Diagnostics.DebuggerHiddenAttribute() */
/** @attribute System.Diagnostics.DebuggerStepThroughAttribute() */
public void AddEventHandler(
   System.Object target,
   System.Delegate handler
);
public System.Reflection.MethodInfo GetAddMethod();
public abstract System.Reflection.MethodInfo GetAddMethod(
   boolean nonPublic
);
public System.Reflection.MethodInfo GetRaiseMethod();
public abstract System.Reflection.MethodInfo GetRaiseMethod(
   boolean nonPublic
);
public System.Reflection.MethodInfo GetRemoveMethod();
public abstract System.Reflection.MethodInfo GetRemoveMethod(
   boolean nonPublic
);
/** @attribute System.Diagnostics.DebuggerStepThroughAttribute() */
/** @attribute System.Diagnostics.DebuggerHiddenAttribute() */
public void RemoveEventHandler(
   System.Object target,
   System.Delegate handler
);
```

Protected Constructors

```
protected EventInfo();
```

FieldAttributes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum FieldAttributes
```

Members

Assembly

FamANDAssem

Family

FamORAssem

FieldAccessMask

HasDefault

HasFieldMarshal

HasFieldRVA

InitOnly

Literal

NotSerialized

PinvokeImpl

Private

PrivateScope

Public

ReservedMask

RTSpecialName

SpecialName

Static

FieldInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDual) */
public abstract class FieldInfo
   extends System.Reflection.MemberInfo
```

Public Properties

```
public abstract System.Reflection.FieldAttributes get_Attributes();
public abstract System.RuntimeFieldHandle get_FieldHandle();
public abstract System.Type get_FieldType();
public boolean get_IsAssembly();
public boolean get_IsFamily();
public boolean get_IsFamilyAndAssembly();
public boolean get_IsFamilyOrAssembly();
public boolean get_IsInitOnly();
public boolean get_IsLiteral();
public boolean get_IsNotSerialized();
public boolean get_IsPinvokeImpl();
public boolean get_IsPrivate();
public boolean get_IsPublic();
public boolean get_IsSpecialName();
public boolean get_IsStatic();
public System.Reflection.MemberTypes get_MemberType();
```

Public Methods

```
public static System.Reflection.FieldInfo GetFieldFromHandle(
   System.RuntimeFieldHandle handle
);
public abstract System.Object GetValue(
   System.Object obj
);
/** @attribute System.Diagnostics.DebuggerHiddenAttribute() */
/** @attribute System.Diagnostics.DebuggerStepThroughAttribute() */
public void SetValue(
   System.Object obj,
   System.Object value
);
public abstract void SetValue(
   System.Object obj,
   System.Object value,
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Globalization.CultureInfo culture
);
```

Protected Constructors

```
protected FieldInfo();
```

ICustomAttributeProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICustomAttributeProvider
```

Public Methods

```
System.Object[] GetCustomAttributes(
   boolean inherit
);
System.Object[] GetCustomAttributes(
   System.Type attributeType,
   boolean inherit
);
boolean IsDefined(
   System.Type attributeType,
   boolean inherit
);
```

InterfaceMapping Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InterfaceMapping
   extends System.ValueType
```

Public Fields

```
public System.Reflection.MethodInfo[] InterfaceMethods;
public System.Type InterfaceType;
public System.Reflection.MethodInfo[] TargetMethods;
public System.Type TargetType;
```

InvalidFilterCriteriaException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class InvalidFilterCriteriaException
   extends System.ApplicationException
```

Public Constructors

```
public InvalidFilterCriteriaException();
public InvalidFilterCriteriaException(
   System.String message
);
public InvalidFilterCriteriaException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected InvalidFilterCriteriaException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

IReflect Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "AFBF15E5-C37C-11d2-B88E-
00A0C9B471B8") */
public interface IReflect
```

Public Properties

```
System.Type get_UnderlyingSystemType();
```

Public Methods

```
System.Reflection.FieldInfo GetField(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
System.Reflection.FieldInfo[] GetFields(
   System.Reflection.BindingFlags bindingAttr
);
System.Reflection.MemberInfo[] GetMember(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
System.Reflection.MemberInfo[] GetMembers(
   System.Reflection.BindingFlags bindingAttr
);
System.Reflection.MethodInfo GetMethod(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
System.Reflection.MethodInfo GetMethod(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
System.Reflection.MethodInfo[] GetMethods(
   System.Reflection.BindingFlags bindingAttr
);
System.Reflection.PropertyInfo[] GetProperties(
   System.Reflection.BindingFlags bindingAttr
);
System.Reflection.PropertyInfo GetProperty(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
System.Reflection.PropertyInfo GetProperty(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Type returnType,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
System.Object InvokeMember(
   System.String name,
```

```
    System.Reflection.BindingFlags invokeAttr,
    System.Reflection.Binder binder,
    System.Object target,
    System.Object[] args,
    System.Reflection.ParameterModifier[] modifiers,
    System.Globalization.CultureInfo culture,
    System.String[] namedParameters
);
```

ManifestResourceInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ManifestResourceInfo
   extends System.Object
```

Public Properties

```
public System.String get_FileName();
public System.Reflection.Assembly get_ReferencedAssembly();
public System.Reflection.ResourceLocation get_ResourceLocation();
```

MemberFilter Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
boolean MemberFilter(
   System.Reflection.MemberInfo m,
   System.Object filterCriteria
);
```

MemberInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDual) */
public abstract class MemberInfo
   extends System.Object
   implements System.Reflection.ICustomAttributeProvider
```

Public Properties

```
public abstract System.Type get_DeclaringType();
public abstract System.Reflection.MemberTypes get_MemberType();
public abstract System.String get_Name();
public abstract System.Type get_ReflectedType();
```

Public Methods

```
public abstract System.Object[] GetCustomAttributes(
   boolean inherit
);
public abstract System.Object[] GetCustomAttributes(
   System.Type attributeType,
   boolean inherit
);
public abstract boolean IsDefined(
   System.Type attributeType,
   boolean inherit
);
```

Protected Constructors

```
protected MemberInfo();
```

MemberTypes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MemberTypes
```

Members

All

Constructor

Custom

Event

Field

Method

NestedType

Property

TypeInfo

MethodAttributes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MethodAttributes
```

Members

Abstract

Assembly

FamANDAssem

Family

FamORAssem

Final

HasSecurity

HideBySig

MemberAccessMask

NewSlot

PinvokeImpl

Private

PrivateScope

Public

RequireSecObject

ReservedMask

ReuseSlot

RTSpecialName

SpecialName

Static

UnmanagedExport

Virtual

VtableLayoutMask

MethodBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDual) */
public abstract class MethodBase
   extends System.Reflection.MemberInfo
```

Public Properties

```
public abstract System.Reflection.MethodAttributes get_Attributes();
public System.Reflection.CallingConventions get_CallingConvention();
public boolean get_IsAbstract();
public boolean get_IsAssembly();
public boolean get_IsConstructor();
public boolean get_IsFamily();
public boolean get_IsFamilyAndAssembly();
public boolean get_IsFamilyOrAssembly();
public boolean get_IsFinal();
public boolean get_IsHideBySig();
public boolean get_IsPrivate();
public boolean get_IsPublic();
public boolean get_IsSpecialName();
public boolean get_IsStatic();
public boolean get_IsVirtual();
public abstract System.RuntimeMethodHandle get_MethodHandle();
```

Public Methods

```
public static System.Reflection.MethodBase GetCurrentMethod();
public static System.Reflection.MethodBase GetMethodFromHandle(
   System.RuntimeMethodHandle handle
);
public abstract System.Reflection.MethodImplAttributes GetMethodImplementationFlags();
public abstract System.Reflection.ParameterInfo[] GetParameters();
/** @attribute System.Diagnostics.DebuggerHiddenAttribute() */
/** @attribute System.Diagnostics.DebuggerStepThroughAttribute() */
public System.Object Invoke(
   System.Object obj,
   System.Object[] parameters
);
public abstract System.Object Invoke(
   System.Object obj,
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Object[] parameters,
   System.Globalization.CultureInfo culture
);
```

Protected Constructors

```
protected MethodBase();
```

MethodImplAttributes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MethodImplAttributes
```

Members

CodeTypeMask

ForwardRef

IL

InternalCall

Managed

ManagedMask

MaxMethodImplVal

Native

NoInlining

OPTIL

PreserveSig

Runtime

Synchronized

Unmanaged

MethodInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDual) */
public abstract class MethodInfo
   extends System.Reflection.MethodBase
```

Public Properties

```
public System.Reflection.MemberTypes get_MemberType();
public abstract System.Type get_ReturnType();
public abstract System.Reflection.ICustomAttributeProvider get_ReturnTypeCustomAttributes();
```

Public Methods

```
public abstract System.Reflection.MethodInfo GetBaseDefinition();
```

Protected Constructors

```
protected MethodInfo();
```

Missing Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Missing
   extends System.Object
```

Public Fields

```
public static final System.Reflection.Missing Value;
```

Module Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Module
   extends System.Object
   implements System.Runtime.Serialization.ISerializable, System.Reflection.ICustomAttributeP
rovider
```

Public Properties

```
public System.Reflection.Assembly get_Assembly();
public System.String get_FullyQualifiedName();
public System.String get_Name();
public System.String get_ScopeName();
```

Public Methods

```
public System.Type[] FindTypes(
   System.Reflection.TypeFilter filter,
   System.Object filterCriteria
);
public System.Object[] GetCustomAttributes(
   boolean inherit
);
public System.Object[] GetCustomAttributes(
   System.Type attributeType,
   boolean inherit
);
public System.Reflection.FieldInfo GetField(
   System.String name
);
public System.Reflection.FieldInfo GetField(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.FieldInfo[] GetFields();
public System.Reflection.MethodInfo GetMethod(
   System.String name
);
public System.Reflection.MethodInfo GetMethod(
   System.String name,
   System.Type[] types
);
public System.Reflection.MethodInfo GetMethod(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Reflection.CallingConventions callConvention,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
public System.Reflection.MethodInfo[] GetMethods();
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public System.Security.Cryptography.X509Certificates.X509Certificate GetSignerCertificate();
```

```
public System.Type GetType(
   System.String className
);
public System.Type GetType(
   System.String className,
   boolean ignoreCase
);
public System.Type GetType(
   System.String className,
   boolean throwOnError,
   boolean ignoreCase
);
public System.Type[] GetTypes();
public boolean IsDefined(
   System.Type attributeType,
   boolean inherit
);
public boolean IsResource();
public System.String ToString();
```

Public Fields

```
public static final System.Reflection.TypeFilter FilterTypeName;
public static final System.Reflection.TypeFilter FilterTypeNameIgnoreCase;
```

Protected Methods

```
protected System.Reflection.MethodInfo GetMethodImpl(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Reflection.CallingConventions callConvention,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
```

ModuleResolveEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
System.Reflection.Module ModuleResolveEventHandler(
   System.Object sender,
   System.ResolveEventArgs e
);
```

ParameterAttributes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum ParameterAttributes
```

Members

HasDefault

HasFieldMarshal

In

Lcid

None

Optional

Out

Reserved3

Reserved4

ReservedMask

Retval

ParameterInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ParameterInfo
   extends System.Object
   implements System.Reflection.ICustomAttributeProvider
```

Public Properties

```
public System.Reflection.ParameterAttributes get_Attributes();
public System.Object get_DefaultValue();
public boolean get_IsIn();
public boolean get_IsLcid();
public boolean get_IsOptional();
public boolean get_IsOut();
public boolean get_IsRetval();
public System.Reflection.MemberInfo get_Member();
public System.String get_Name();
public System.Type get_ParameterType();
public int get_Position();
```

Public Methods

```
public System.Object[] GetCustomAttributes(
   boolean inherit
);
public System.Object[] GetCustomAttributes(
   System.Type attributeType,
   boolean inherit
);
public boolean IsDefined(
   System.Type attributeType,
   boolean inherit
);
```

Protected Constructors

```
protected ParameterInfo();
```

Protected Fields

```
protected System.Reflection.ParameterAttributes AttrsImpl;
protected System.Type ClassImpl;
protected System.Object DefaultValueImpl;
protected System.Reflection.MemberInfo MemberImpl;
protected System.String NameImpl;
protected int PositionImpl;
```

ParameterModifier Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ParameterModifier
   extends System.ValueType
```

Public Constructors

```
public ParameterModifier(
   int paramaterCount
);
```

Public Properties

```
public boolean get_Item(
   int index
);
public void set_Item(
   int index,
   boolean value
);
```

Pointer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public class Pointer
   extends System.Object
   implements System.Runtime.Serialization.ISerializable
```

Public Methods

```
public static System.Object Box(
   void* ptr,
   System.Type type
);
public static void* Unbox(
   System.Object ptr
);
```

PropertyAttributes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum PropertyAttributes
```

Members

HasDefault

None

Reserved2

Reserved3

Reserved4

ReservedMask

RTSpecialName

SpecialName

PropertyInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDual) */
public abstract class PropertyInfo
   extends System.Reflection.MemberInfo
```

Public Properties

```
public abstract System.Reflection.PropertyAttributes get_Attributes();
public abstract boolean get_CanRead();
public abstract boolean get_CanWrite();
public boolean get_IsSpecialName();
public System.Reflection.MemberTypes get_MemberType();
public abstract System.Type get_PropertyType();
```

Public Methods

```
public System.Reflection.MethodInfo[] GetAccessors();
public abstract System.Reflection.MethodInfo[] GetAccessors(
   boolean nonPublic
);
public System.Reflection.MethodInfo GetGetMethod();
public abstract System.Reflection.MethodInfo GetGetMethod(
   boolean nonPublic
);
public abstract System.Reflection.ParameterInfo[] GetIndexParameters();
public System.Reflection.MethodInfo GetSetMethod();
public abstract System.Reflection.MethodInfo GetSetMethod(
   boolean nonPublic
);
/** @attribute System.Diagnostics.DebuggerHiddenAttribute() */
/** @attribute System.Diagnostics.DebuggerStepThroughAttribute() */
public System.Object GetValue(
   System.Object obj,
   System.Object[] index
);
public abstract System.Object GetValue(
   System.Object obj,
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Object[] index,
   System.Globalization.CultureInfo culture
);
/** @attribute System.Diagnostics.DebuggerHiddenAttribute() */
/** @attribute System.Diagnostics.DebuggerStepThroughAttribute() */
public void SetValue(
   System.Object obj,
   System.Object value,
   System.Object[] index
);
public abstract void SetValue(
   System.Object obj,
   System.Object value,
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Object[] index,
```

```
    System.Globalization.CultureInfo culture
);
```

Protected Constructors

```
protected PropertyInfo();
```

ReflectionTypeLoadException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ReflectionTypeLoadException
   extends System.SystemException
```

Public Constructors

```
public ReflectionTypeLoadException(
   System.Type[] classes,
   System.Exception[] exceptions
);
public ReflectionTypeLoadException(
   System.Type[] classes,
   System.Exception[] exceptions,
   System.String message
);
```

Public Properties

```
public System.Exception[] get_LoaderExceptions();
public System.Type[] get_Types();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ResourceAttributes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum ResourceAttributes
```

Members

Private

Public

ResourceLocation Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum ResourceLocation
```

Members

ContainedInAnotherAssembly

ContainedInManifestFile

Embedded

StrongNameKeyPair Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StrongNameKeyPair
   extends System.Object
```

Public Constructors

```
public StrongNameKeyPair(
   ubyte[] keyPairArray
);
public StrongNameKeyPair(
   System.IO.FileStream keyPairFile
);
public StrongNameKeyPair(
   System.String keyPairContainer
);
```

Public Properties

```
public ubyte[] get_PublicKey();
```

TargetException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class TargetException
   extends System.ApplicationException
```

Public Constructors

```
public TargetException();
public TargetException(
   System.String message
);
public TargetException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected TargetException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

TargetInvocationException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class TargetInvocationException
   extends System.ApplicationException
```

Public Constructors

```
public TargetInvocationException(
   System.Exception inner
);
public TargetInvocationException(
   System.String message,
   System.Exception inner
);
```

TargetParameterCountException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class TargetParameterCountException
   extends System.ApplicationException
```

Public Constructors

```
public TargetParameterCountException();
public TargetParameterCountException(
   System.String message
);
public TargetParameterCountException(
   System.String message,
   System.Exception inner
);
```

TypeAttributes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum TypeAttributes
```

Members

Abstract

AnsiClass

AutoClass

AutoLayout

BeforeFieldInit

Class

ClassSemanticsMask

ExplicitLayout

HasSecurity

Import

Interface

LayoutMask

NestedAssembly

NestedFamANDAssem

NestedFamily

NestedFamORAssem

NestedPrivate

NestedPublic

NotPublic

Public

ReservedMask

RTSpecialName

Sealed

SequentialLayout

Serializable

SpecialName

StringFormatMask

UnicodeClass

VisibilityMask

TypeDelegator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class TypeDelegator
   extends System.Type
```

Public Constructors

```
public TypeDelegator(
   System.Type delegatingType
);
```

Public Properties

```
public System.Reflection.Assembly get_Assembly();
public System.String get_AssemblyQualifiedName();
public System.Type get_BaseType();
public System.String get_FullName();
public System.Guid get_GUID();
public System.Reflection.Module get_Module();
public System.String get_Name();
public System.String get_Namespace();
public System.RuntimeTypeHandle get_TypeHandle();
public System.Type get_UnderlyingSystemType();
```

Public Methods

```
public System.Reflection.ConstructorInfo[] GetConstructors(
   System.Reflection.BindingFlags bindingAttr
);
public System.Object[] GetCustomAttributes(
   boolean inherit
);
public System.Object[] GetCustomAttributes(
   System.Type attributeType,
   boolean inherit
);
public System.Type GetElementType();
public System.Reflection.EventInfo GetEvent(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.EventInfo[] GetEvents();
public System.Reflection.EventInfo[] GetEvents(
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.FieldInfo GetField(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.FieldInfo[] GetFields(
   System.Reflection.BindingFlags bindingAttr
);
public System.Type GetInterface(
   System.String name,
```

```
   boolean ignoreCase
);
public System.Reflection.InterfaceMapping GetInterfaceMap(
   System.Type interfaceType
);
public System.Type[] GetInterfaces();
public System.Reflection.MemberInfo[] GetMember(
   System.String name,
   System.Reflection.MemberTypes type,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.MemberInfo[] GetMembers(
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.MethodInfo[] GetMethods(
   System.Reflection.BindingFlags bindingAttr
);
public System.Type GetNestedType(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Type[] GetNestedTypes(
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.PropertyInfo[] GetProperties(
   System.Reflection.BindingFlags bindingAttr
);
public System.Object InvokeMember(
   System.String name,
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Object target,
   System.Object[] args,
   System.Reflection.ParameterModifier[] modifiers,
   System.Globalization.CultureInfo culture,
   System.String[] namedParameters
);
public boolean IsDefined(
   System.Type attributeType,
   boolean inherit
);
```

Protected Constructors

```
protected TypeDelegator();
```

Protected Methods

```
protected System.Reflection.TypeAttributes GetAttributeFlagsImpl();
protected System.Reflection.ConstructorInfo GetConstructorImpl(
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Reflection.CallingConventions callConvention,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
protected System.Reflection.MethodInfo GetMethodImpl(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Reflection.CallingConventions callConvention,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
```

```
protected System.Reflection.PropertyInfo GetPropertyImpl(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Type returnType,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
protected boolean HasElementTypeImpl();
protected boolean IsArrayImpl();
protected boolean IsByRefImpl();
protected boolean IsCOMObjectImpl();
protected boolean IsPointerImpl();
protected boolean IsPrimitiveImpl();
protected boolean IsValueTypeImpl();
```

Protected Fields

```
protected System.Type typeImpl;
```

TypeFilter Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
boolean TypeFilter(
   System.Type m,
   System.Object filterCriteria
);
```

System.Reflection.Emit Namespace (Visual J# Syntax)

Classes

AssemblyBuilder

ConstructorBuilder

CustomAttributeBuilder

EnumBuilder

EventBuilder

FieldBuilder

ILGenerator

LocalBuilder

MethodBuilder

MethodRental

ModuleBuilder

OpCodes

ParameterBuilder

PropertyBuilder

SignatureHelper

TypeBuilder

UnmanagedMarshal

Structures

EventToken

FieldToken

Label

MethodToken

OpCode

ParameterToken

PropertyToken

SignatureToken

StringToken

TypeToken

Enumerations

AssemblyBuilderAccess

FlowControl

OpCodeType

OperandType

PackingSize

PEFileKinds

StackBehaviour

AssemblyBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AssemblyBuilder
   extends System.Reflection.Assembly
```

Public Properties

```
public System.String get_CodeBase();
public System.Reflection.MethodInfo get_EntryPoint();
public System.String get_Location();
```

Public Methods

```
public void AddResourceFile(
   System.String name,
   System.String fileName
);
public void AddResourceFile(
   System.String name,
   System.String fileName,
   System.Reflection.ResourceAttributes attribute
);
public System.Reflection.Emit.ModuleBuilder DefineDynamicModule(
   System.String name
);
public System.Reflection.Emit.ModuleBuilder DefineDynamicModule(
   System.String name,
   boolean emitSymbolInfo
);
public System.Reflection.Emit.ModuleBuilder DefineDynamicModule(
   System.String name,
   System.String fileName
);
public System.Reflection.Emit.ModuleBuilder DefineDynamicModule(
   System.String name,
   System.String fileName,
   boolean emitSymbolInfo
);
public System.Resources.IResourceWriter DefineResource(
   System.String name,
   System.String description,
   System.String fileName
);
public System.Resources.IResourceWriter DefineResource(
   System.String name,
   System.String description,
   System.String fileName,
   System.Reflection.ResourceAttributes attribute
);
public void DefineUnmanagedResource(
   ubyte[] resource
);
public void DefineUnmanagedResource(
   System.String resourceFileName
);
public void DefineVersionInfoResource();
public void DefineVersionInfoResource(
```

```
    System.String product,
    System.String productVersion,
    System.String company,
    System.String copyright,
    System.String trademark
);
public System.Reflection.Emit.ModuleBuilder GetDynamicModule(
    System.String name
);
public System.Type[] GetExportedTypes();
public System.IO.FileStream GetFile(
    System.String name
);
public System.IO.FileStream[] GetFiles(
    boolean getResourceModules
);
public System.Reflection.ManifestResourceInfo GetManifestResourceInfo(
    System.String resourceName
);
public System.String[] GetManifestResourceNames();
public System.IO.Stream GetManifestResourceStream(
    System.String name
);
public System.IO.Stream GetManifestResourceStream(
    System.Type type,
    System.String name
);
public void Save(
    System.String assemblyFileName
);
public void SetCustomAttribute(
    System.Reflection.Emit.CustomAttributeBuilder customBuilder
);
public void SetCustomAttribute(
    System.Reflection.ConstructorInfo con,
    ubyte[] binaryAttribute
);
public void SetEntryPoint(
    System.Reflection.MethodInfo entryMethod
);
public void SetEntryPoint(
    System.Reflection.MethodInfo entryMethod,
    System.Reflection.Emit.PEFileKinds fileKind
);
```

AssemblyBuilderAccess Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum AssemblyBuilderAccess
```

Members

Run

RunAndSave

Save

ConstructorBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ConstructorBuilder
   extends System.Reflection.ConstructorInfo
```

Public Properties

```
public System.Reflection.MethodAttributes get_Attributes();
public System.Type get_DeclaringType();
public boolean get_InitLocals();
public void set_InitLocals(
   boolean value
);
public System.RuntimeMethodHandle get_MethodHandle();
public System.String get_Name();
public System.Type get_ReflectedType();
public System.Type get_ReturnType();
public System.String get_Signature();
```

Public Methods

```
public void AddDeclarativeSecurity(
   System.Security.Permissions.SecurityAction action,
   System.Security.PermissionSet pset
);
public System.Reflection.Emit.ParameterBuilder DefineParameter(
   int iSequence,
   System.Reflection.ParameterAttributes attributes,
   System.String strParamName
);
public System.Object[] GetCustomAttributes(
   boolean inherit
);
public System.Object[] GetCustomAttributes(
   System.Type attributeType,
   boolean inherit
);
public System.Reflection.Emit.ILGenerator GetILGenerator();
public System.Reflection.MethodImplAttributes GetMethodImplementationFlags();
public System.Reflection.Module GetModule();
public System.Reflection.ParameterInfo[] GetParameters();
public System.Reflection.Emit.MethodToken GetToken();
public System.Object Invoke(
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Object[] parameters,
   System.Globalization.CultureInfo culture
);
public System.Object Invoke(
   System.Object obj,
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Object[] parameters,
   System.Globalization.CultureInfo culture
);
public boolean IsDefined(
   System.Type attributeType,
```

```
      boolean inherit
   );
   public void SetCustomAttribute(
      System.Reflection.Emit.CustomAttributeBuilder customBuilder
   );
   public void SetCustomAttribute(
      System.Reflection.ConstructorInfo con,
      ubyte[] binaryAttribute
   );
   public void SetImplementationFlags(
      System.Reflection.MethodImplAttributes attributes
   );
   public void SetSymCustomAttribute(
      System.String name,
      ubyte[] data
   );
   public System.String ToString();
```

CustomAttributeBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CustomAttributeBuilder
   extends System.Object
```

Public Constructors

```
public CustomAttributeBuilder(
   System.Reflection.ConstructorInfo con,
   System.Object[] constructorArgs
);
public CustomAttributeBuilder(
   System.Reflection.ConstructorInfo con,
   System.Object[] constructorArgs,
   System.Reflection.FieldInfo[] namedFields,
   System.Object[] fieldValues
);
public CustomAttributeBuilder(
   System.Reflection.ConstructorInfo con,
   System.Object[] constructorArgs,
   System.Reflection.PropertyInfo[] namedProperties,
   System.Object[] propertyValues
);
public CustomAttributeBuilder(
   System.Reflection.ConstructorInfo con,
   System.Object[] constructorArgs,
   System.Reflection.PropertyInfo[] namedProperties,
   System.Object[] propertyValues,
   System.Reflection.FieldInfo[] namedFields,
   System.Object[] fieldValues
);
```

EnumBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EnumBuilder
   extends System.Type
```

Public Properties

```
public System.Reflection.Assembly get_Assembly();
public System.String get_AssemblyQualifiedName();
public System.Type get_BaseType();
public System.Type get_DeclaringType();
public System.String get_FullName();
public System.Guid get_GUID();
public System.Reflection.Module get_Module();
public System.String get_Name();
public System.String get_Namespace();
public System.Type get_ReflectedType();
public System.RuntimeTypeHandle get_TypeHandle();
public System.Reflection.Emit.TypeToken get_TypeToken();
public System.Reflection.Emit.FieldBuilder get_UnderlyingField();
public System.Type get_UnderlyingSystemType();
```

Public Methods

```
public System.Type CreateType();
public System.Reflection.Emit.FieldBuilder DefineLiteral(
   System.String literalName,
   System.Object literalValue
);
public System.Reflection.ConstructorInfo[] GetConstructors(
   System.Reflection.BindingFlags bindingAttr
);
public System.Object[] GetCustomAttributes(
   boolean inherit
);
public System.Object[] GetCustomAttributes(
   System.Type attributeType,
   boolean inherit
);
public System.Type GetElementType();
public System.Reflection.EventInfo GetEvent(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.EventInfo[] GetEvents();
public System.Reflection.EventInfo[] GetEvents(
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.FieldInfo GetField(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.FieldInfo[] GetFields(
   System.Reflection.BindingFlags bindingAttr
);
public System.Type GetInterface(
   System.String name,
```

```
   boolean ignoreCase
);
public System.Reflection.InterfaceMapping GetInterfaceMap(
   System.Type interfaceType
);
public System.Type[] GetInterfaces();
public System.Reflection.MemberInfo[] GetMember(
   System.String name,
   System.Reflection.MemberTypes type,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.MemberInfo[] GetMembers(
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.MethodInfo[] GetMethods(
   System.Reflection.BindingFlags bindingAttr
);
public System.Type GetNestedType(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Type[] GetNestedTypes(
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.PropertyInfo[] GetProperties(
   System.Reflection.BindingFlags bindingAttr
);
public System.Object InvokeMember(
   System.String name,
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Object target,
   System.Object[] args,
   System.Reflection.ParameterModifier[] modifiers,
   System.Globalization.CultureInfo culture,
   System.String[] namedParameters
);
public boolean IsDefined(
   System.Type attributeType,
   boolean inherit
);
public void SetCustomAttribute(
   System.Reflection.Emit.CustomAttributeBuilder customBuilder
);
public void SetCustomAttribute(
   System.Reflection.ConstructorInfo con,
   ubyte[] binaryAttribute
);
```

Protected Methods

```
protected System.Reflection.TypeAttributes GetAttributeFlagsImpl();
protected System.Reflection.ConstructorInfo GetConstructorImpl(
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Reflection.CallingConventions callConvention,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
protected System.Reflection.MethodInfo GetMethodImpl(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Reflection.CallingConventions callConvention,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
```

```
);
protected System.Reflection.PropertyInfo GetPropertyImpl(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Type returnType,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
protected boolean HasElementTypeImpl();
protected boolean IsArrayImpl();
protected boolean IsByRefImpl();
protected boolean IsCOMObjectImpl();
protected boolean IsPointerImpl();
protected boolean IsPrimitiveImpl();
protected boolean IsValueTypeImpl();
```

EventBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EventBuilder
   extends System.Object
```

Public Methods

```
public void AddOtherMethod(
   System.Reflection.Emit.MethodBuilder mdBuilder
);
public System.Reflection.Emit.EventToken GetEventToken();
public void SetAddOnMethod(
   System.Reflection.Emit.MethodBuilder mdBuilder
);
public void SetCustomAttribute(
   System.Reflection.Emit.CustomAttributeBuilder customBuilder
);
public void SetCustomAttribute(
   System.Reflection.ConstructorInfo con,
   ubyte[] binaryAttribute
);
public void SetRaiseMethod(
   System.Reflection.Emit.MethodBuilder mdBuilder
);
public void SetRemoveOnMethod(
   System.Reflection.Emit.MethodBuilder mdBuilder
);
```

EventToken Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class EventToken
   extends System.ValueType
```

Public Properties

```
public int get_Token();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

Public Fields

```
public static final System.Reflection.Emit.EventToken Empty;
```

FieldBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FieldBuilder
   extends System.Reflection.FieldInfo
```

Public Properties

```
public System.Reflection.FieldAttributes get_Attributes();
public System.Type get_DeclaringType();
public System.RuntimeFieldHandle get_FieldHandle();
public System.Type get_FieldType();
public System.String get_Name();
public System.Type get_ReflectedType();
```

Public Methods

```
public System.Object[] GetCustomAttributes(
   boolean inherit
);
public System.Object[] GetCustomAttributes(
   System.Type attributeType,
   boolean inherit
);
public System.Reflection.Emit.FieldToken GetToken();
public System.Object GetValue(
   System.Object obj
);
public boolean IsDefined(
   System.Type attributeType,
   boolean inherit
);
public void SetConstant(
   System.Object defaultValue
);
public void SetCustomAttribute(
   System.Reflection.Emit.CustomAttributeBuilder customBuilder
);
public void SetCustomAttribute(
   System.Reflection.ConstructorInfo con,
   ubyte[] binaryAttribute
);
public void SetMarshal(
   System.Reflection.Emit.UnmanagedMarshal unmanagedMarshal
);
public void SetOffset(
   int iOffset
);
public void SetValue(
   System.Object obj,
   System.Object val,
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Globalization.CultureInfo culture
);
```

FieldToken Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class FieldToken
   extends System.ValueType
```

Public Properties

```
public int get_Token();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

Public Fields

```
public static final System.Reflection.Emit.FieldToken Empty;
```

FlowControl Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum FlowControl
```

Members

Branch

Break

Call

Cond_Branch

Meta

Next

Phi

Return

Throw

ILGenerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ILGenerator
   extends System.Object
```

Public Methods

```
public void BeginCatchBlock(
   System.Type exceptionType
);
public void BeginExceptFilterBlock();
public System.Reflection.Emit.Label BeginExceptionBlock();
public void BeginFaultBlock();
public void BeginFinallyBlock();
public void BeginScope();
public System.Reflection.Emit.LocalBuilder DeclareLocal(
   System.Type localType
);
public System.Reflection.Emit.Label DefineLabel();
public void Emit(
   System.Reflection.Emit.OpCode opcode
);
public void Emit(
   System.Reflection.Emit.OpCode opcode,
   ubyte arg
);
public void Emit(
   System.Reflection.Emit.OpCode opcode,
   double arg
);
public void Emit(
   System.Reflection.Emit.OpCode opcode,
   short arg
);
public void Emit(
   System.Reflection.Emit.OpCode opcode,
   int arg
);
public void Emit(
   System.Reflection.Emit.OpCode opcode,
   long arg
);
public void Emit(
   System.Reflection.Emit.OpCode opcode,
   float arg
);
public void Emit(
   System.Reflection.Emit.OpCode opcode,
   System.Reflection.ConstructorInfo con
);
public void Emit(
   System.Reflection.Emit.OpCode opcode,
   System.Reflection.Emit.Label label
);
public void Emit(
   System.Reflection.Emit.OpCode opcode,
   System.Reflection.Emit.Label[] labels
);
public void Emit(
```

```
   System.Reflection.Emit.OpCode opcode,
   System.Reflection.Emit.LocalBuilder local
);
public void Emit(
   System.Reflection.Emit.OpCode opcode,
   System.Reflection.Emit.SignatureHelper signature
);
public void Emit(
   System.Reflection.Emit.OpCode opcode,
   System.Reflection.FieldInfo field
);
public void Emit(
   System.Reflection.Emit.OpCode opcode,
   System.Reflection.MethodInfo meth
);
public void Emit(
   System.Reflection.Emit.OpCode opcode,
   System.String str
);
public void Emit(
   System.Reflection.Emit.OpCode opcode,
   System.Type cls
);
public void EmitCall(
   System.Reflection.Emit.OpCode opcode,
   System.Reflection.MethodInfo methodInfo,
   System.Type[] optionalParameterTypes
);
public void EmitCalli(
   System.Reflection.Emit.OpCode opcode,
   System.Runtime.InteropServices.CallingConvention unmanagedCallConv,
   System.Type returnType,
   System.Type[] parameterTypes
);
public void EmitCalli(
   System.Reflection.Emit.OpCode opcode,
   System.Reflection.CallingConventions callingConvention,
   System.Type returnType,
   System.Type[] parameterTypes,
   System.Type[] optionalParameterTypes
);
public void EmitWriteLine(
   System.Reflection.Emit.LocalBuilder localBuilder
);
public void EmitWriteLine(
   System.Reflection.FieldInfo fld
);
public void EmitWriteLine(
   System.String value
);
public void EndExceptionBlock();
public void EndScope();
public void MarkLabel(
   System.Reflection.Emit.Label loc
);
public void MarkSequencePoint(
   System.Diagnostics.SymbolStore.ISymbolDocumentWriter document,
   int startLine,
   int startColumn,
   int endLine,
   int endColumn
);
public void ThrowException(
   System.Type excType
);
public void UsingNamespace(
   System.String usingNamespace
);
```

Label Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Label
   extends System.ValueType
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

LocalBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LocalBuilder
   extends System.Object
```

Public Properties

```
public System.Type get_LocalType();
```

Public Methods

```
public void SetLocalSymInfo(
   System.String name
);
public void SetLocalSymInfo(
   System.String name,
   int startOffset,
   int endOffset
);
```

MethodBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MethodBuilder
   extends System.Reflection.MethodInfo
```

Public Properties

```
public System.Reflection.MethodAttributes get_Attributes();
public System.Reflection.CallingConventions get_CallingConvention();
public System.Type get_DeclaringType();
public boolean get_InitLocals();
public void set_InitLocals(
   boolean value
);
public System.RuntimeMethodHandle get_MethodHandle();
public System.String get_Name();
public System.Type get_ReflectedType();
public System.Type get_ReturnType();
public System.Reflection.ICustomAttributeProvider get_ReturnTypeCustomAttributes();
public System.String get_Signature();
```

Public Methods

```
public void AddDeclarativeSecurity(
   System.Security.Permissions.SecurityAction action,
   System.Security.PermissionSet pset
);
public void CreateMethodBody(
   ubyte[] il,
   int count
);
public System.Reflection.Emit.ParameterBuilder DefineParameter(
   int position,
   System.Reflection.ParameterAttributes attributes,
   System.String strParamName
);
public boolean Equals(
   System.Object obj
);
public System.Reflection.MethodInfo GetBaseDefinition();
public System.Object[] GetCustomAttributes(
   boolean inherit
);
public System.Object[] GetCustomAttributes(
   System.Type attributeType,
   boolean inherit
);
public int GetHashCode();
public System.Reflection.Emit.ILGenerator GetILGenerator();
public System.Reflection.Emit.ILGenerator GetILGenerator(
   int size
);
public System.Reflection.MethodImplAttributes GetMethodImplementationFlags();
public System.Reflection.Module GetModule();
public System.Reflection.ParameterInfo[] GetParameters();
public System.Reflection.Emit.MethodToken GetToken();
public System.Object Invoke(
```

```
      System.Object obj,
      System.Reflection.BindingFlags invokeAttr,
      System.Reflection.Binder binder,
      System.Object[] parameters,
      System.Globalization.CultureInfo culture
   );
   public boolean IsDefined(
      System.Type attributeType,
      boolean inherit
   );
   public void SetCustomAttribute(
      System.Reflection.Emit.CustomAttributeBuilder customBuilder
   );
   public void SetCustomAttribute(
      System.Reflection.ConstructorInfo con,
      ubyte[] binaryAttribute
   );
   public void SetImplementationFlags(
      System.Reflection.MethodImplAttributes attributes
   );
   public void SetMarshal(
      System.Reflection.Emit.UnmanagedMarshal unmanagedMarshal
   );
   public void SetSymCustomAttribute(
      System.String name,
      ubyte[] data
   );
   public System.String ToString();
```

MethodRental Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MethodRental
   extends System.Object
```

Public Methods

```
public static void SwapMethodBody(
   System.Type cls,
   int methodtoken,
   System.IntPtr rgIL,
   int methodSize,
   int flags
);
```

Public Fields

```
public static int JitImmediate;
public static int JitOnDemand;
```

MethodToken Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class MethodToken
   extends System.ValueType
```

Public Properties

```
public int get_Token();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

Public Fields

```
public static final System.Reflection.Emit.MethodToken Empty;
```

ModuleBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ModuleBuilder
   extends System.Reflection.Module
```

Public Properties

```
public System.String get_FullyQualifiedName();
```

Public Methods

```
public void CreateGlobalFunctions();
public System.Diagnostics.SymbolStore.ISymbolDocumentWriter DefineDocument(
   System.String url,
   System.Guid language,
   System.Guid languageVendor,
   System.Guid documentType
);
public System.Reflection.Emit.EnumBuilder DefineEnum(
   System.String name,
   System.Reflection.TypeAttributes visibility,
   System.Type underlyingType
);
public System.Reflection.Emit.MethodBuilder DefineGlobalMethod(
   System.String name,
   System.Reflection.MethodAttributes attributes,
   System.Type returnType,
   System.Type[] parameterTypes
);
public System.Reflection.Emit.MethodBuilder DefineGlobalMethod(
   System.String name,
   System.Reflection.MethodAttributes attributes,
   System.Reflection.CallingConventions callingConvention,
   System.Type returnType,
   System.Type[] parameterTypes
);
public System.Reflection.Emit.FieldBuilder DefineInitializedData(
   System.String name,
   ubyte[] data,
   System.Reflection.FieldAttributes attributes
);
public System.Reflection.Emit.MethodBuilder DefinePInvokeMethod(
   System.String name,
   System.String dllName,
   System.Reflection.MethodAttributes attributes,
   System.Reflection.CallingConventions callingConvention,
   System.Type returnType,
   System.Type[] parameterTypes,
   System.Runtime.InteropServices.CallingConvention nativeCallConv,
   System.Runtime.InteropServices.CharSet nativeCharSet
);
public System.Reflection.Emit.MethodBuilder DefinePInvokeMethod(
   System.String name,
   System.String dllName,
   System.String entryName,
   System.Reflection.MethodAttributes attributes,
   System.Reflection.CallingConventions callingConvention,
```

```
      System.Type returnType,
      System.Type[] parameterTypes,
      System.Runtime.InteropServices.CallingConvention nativeCallConv,
      System.Runtime.InteropServices.CharSet nativeCharSet
   );
   public System.Resources.IResourceWriter DefineResource(
      System.String name,
      System.String description
   );
   public System.Resources.IResourceWriter DefineResource(
      System.String name,
      System.String description,
      System.Reflection.ResourceAttributes attribute
   );
   public System.Reflection.Emit.TypeBuilder DefineType(
      System.String name
   );
   public System.Reflection.Emit.TypeBuilder DefineType(
      System.String name,
      System.Reflection.TypeAttributes attr
   );
   public System.Reflection.Emit.TypeBuilder DefineType(
      System.String name,
      System.Reflection.TypeAttributes attr,
      System.Type parent
   );
   public System.Reflection.Emit.TypeBuilder DefineType(
      System.String name,
      System.Reflection.TypeAttributes attr,
      System.Type parent,
      int typesize
   );
   public System.Reflection.Emit.TypeBuilder DefineType(
      System.String name,
      System.Reflection.TypeAttributes attr,
      System.Type parent,
      System.Reflection.Emit.PackingSize packsize
   );
   public System.Reflection.Emit.TypeBuilder DefineType(
      System.String name,
      System.Reflection.TypeAttributes attr,
      System.Type parent,
      System.Type[] interfaces
   );
   public System.Reflection.Emit.TypeBuilder DefineType(
      System.String name,
      System.Reflection.TypeAttributes attr,
      System.Type parent,
      System.Reflection.Emit.PackingSize packingSize,
      int typesize
   );
   public System.Reflection.Emit.FieldBuilder DefineUninitializedData(
      System.String name,
      int size,
      System.Reflection.FieldAttributes attributes
   );
   public void DefineUnmanagedResource(
      ubyte[] resource
   );
   public void DefineUnmanagedResource(
      System.String resourceFileName
   );
   public System.Reflection.MethodInfo GetArrayMethod(
      System.Type arrayClass,
      System.String methodName,
      System.Reflection.CallingConventions callingConvention,
      System.Type returnType,
      System.Type[] parameterTypes
```

```
);
public System.Reflection.Emit.MethodToken GetArrayMethodToken(
   System.Type arrayClass,
   System.String methodName,
   System.Reflection.CallingConventions callingConvention,
   System.Type returnType,
   System.Type[] parameterTypes
);
public System.Reflection.Emit.MethodToken GetConstructorToken(
   System.Reflection.ConstructorInfo con
);
public System.Reflection.Emit.FieldToken GetFieldToken(
   System.Reflection.FieldInfo field
);
public System.Reflection.Emit.MethodToken GetMethodToken(
   System.Reflection.MethodInfo method
);
public System.Reflection.Emit.SignatureToken GetSignatureToken(
   System.Reflection.Emit.SignatureHelper sigHelper
);
public System.Reflection.Emit.SignatureToken GetSignatureToken(
   ubyte[] sigBytes,
   int sigLength
);
public System.Reflection.Emit.StringToken GetStringConstant(
   System.String str
);
public System.Diagnostics.SymbolStore.ISymbolWriter GetSymWriter();
public System.Type GetType(
   System.String className
);
public System.Type GetType(
   System.String className,
   boolean ignoreCase
);
public System.Type GetType(
   System.String className,
   boolean throwOnError,
   boolean ignoreCase
);
public System.Type[] GetTypes();
public System.Reflection.Emit.TypeToken GetTypeToken(
   System.String name
);
public System.Reflection.Emit.TypeToken GetTypeToken(
   System.Type type
);
public boolean IsTransient();
public void SetCustomAttribute(
   System.Reflection.Emit.CustomAttributeBuilder customBuilder
);
public void SetCustomAttribute(
   System.Reflection.ConstructorInfo con,
   ubyte[] binaryAttribute
);
public void SetSymCustomAttribute(
   System.String name,
   ubyte[] data
);
public void SetUserEntryPoint(
   System.Reflection.MethodInfo entryPoint
);
```

OpCode Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class OpCode
   extends System.ValueType
```

Public Properties

```
public System.Reflection.Emit.FlowControl get_FlowControl();
public System.String get_Name();
public System.Reflection.Emit.OpCodeType get_OpCodeType();
public System.Reflection.Emit.OperandType get_OperandType();
public int get_Size();
public System.Reflection.Emit.StackBehaviour get_StackBehaviourPop();
public System.Reflection.Emit.StackBehaviour get_StackBehaviourPush();
public short get_Value();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public System.String ToString();
```

OpCodes Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class OpCodes
   extends System.Object
```

Public Methods

```
public static boolean TakesSingleByteArgument(
   System.Reflection.Emit.OpCode inst
);
```

Public Fields

```
public static final System.Reflection.Emit.OpCode Add;
public static final System.Reflection.Emit.OpCode Add_Ovf;
public static final System.Reflection.Emit.OpCode Add_Ovf_Un;
public static final System.Reflection.Emit.OpCode And;
public static final System.Reflection.Emit.OpCode Arglist;
public static final System.Reflection.Emit.OpCode Beq;
public static final System.Reflection.Emit.OpCode Beq_S;
public static final System.Reflection.Emit.OpCode Bge;
public static final System.Reflection.Emit.OpCode Bge_S;
public static final System.Reflection.Emit.OpCode Bge_Un;
public static final System.Reflection.Emit.OpCode Bge_Un_S;
public static final System.Reflection.Emit.OpCode Bgt;
public static final System.Reflection.Emit.OpCode Bgt_S;
public static final System.Reflection.Emit.OpCode Bgt_Un;
public static final System.Reflection.Emit.OpCode Bgt_Un_S;
public static final System.Reflection.Emit.OpCode Ble;
public static final System.Reflection.Emit.OpCode Ble_S;
public static final System.Reflection.Emit.OpCode Ble_Un;
public static final System.Reflection.Emit.OpCode Ble_Un_S;
public static final System.Reflection.Emit.OpCode Blt;
public static final System.Reflection.Emit.OpCode Blt_S;
public static final System.Reflection.Emit.OpCode Blt_Un;
public static final System.Reflection.Emit.OpCode Blt_Un_S;
public static final System.Reflection.Emit.OpCode Bne_Un;
public static final System.Reflection.Emit.OpCode Bne_Un_S;
public static final System.Reflection.Emit.OpCode Box;
public static final System.Reflection.Emit.OpCode Br;
public static final System.Reflection.Emit.OpCode Br_S;
public static final System.Reflection.Emit.OpCode Break;
public static final System.Reflection.Emit.OpCode Brfalse;
public static final System.Reflection.Emit.OpCode Brfalse_S;
public static final System.Reflection.Emit.OpCode Brtrue;
public static final System.Reflection.Emit.OpCode Brtrue_S;
public static final System.Reflection.Emit.OpCode Call;
public static final System.Reflection.Emit.OpCode Calli;
public static final System.Reflection.Emit.OpCode Callvirt;
public static final System.Reflection.Emit.OpCode Castclass;
public static final System.Reflection.Emit.OpCode Ceq;
public static final System.Reflection.Emit.OpCode Cgt;
public static final System.Reflection.Emit.OpCode Cgt_Un;
public static final System.Reflection.Emit.OpCode Ckfinite;
public static final System.Reflection.Emit.OpCode Clt;
public static final System.Reflection.Emit.OpCode Clt_Un;
public static final System.Reflection.Emit.OpCode Conv_I;
```

```
public static final System.Reflection.Emit.OpCode Conv_I1;
public static final System.Reflection.Emit.OpCode Conv_I2;
public static final System.Reflection.Emit.OpCode Conv_I4;
public static final System.Reflection.Emit.OpCode Conv_I8;
public static final System.Reflection.Emit.OpCode Conv_Ovf_I;
public static final System.Reflection.Emit.OpCode Conv_Ovf_I_Un;
public static final System.Reflection.Emit.OpCode Conv_Ovf_I1;
public static final System.Reflection.Emit.OpCode Conv_Ovf_I1_Un;
public static final System.Reflection.Emit.OpCode Conv_Ovf_I2;
public static final System.Reflection.Emit.OpCode Conv_Ovf_I2_Un;
public static final System.Reflection.Emit.OpCode Conv_Ovf_I4;
public static final System.Reflection.Emit.OpCode Conv_Ovf_I4_Un;
public static final System.Reflection.Emit.OpCode Conv_Ovf_I8;
public static final System.Reflection.Emit.OpCode Conv_Ovf_I8_Un;
public static final System.Reflection.Emit.OpCode Conv_Ovf_U;
public static final System.Reflection.Emit.OpCode Conv_Ovf_U_Un;
public static final System.Reflection.Emit.OpCode Conv_Ovf_U1;
public static final System.Reflection.Emit.OpCode Conv_Ovf_U1_Un;
public static final System.Reflection.Emit.OpCode Conv_Ovf_U2;
public static final System.Reflection.Emit.OpCode Conv_Ovf_U2_Un;
public static final System.Reflection.Emit.OpCode Conv_Ovf_U4;
public static final System.Reflection.Emit.OpCode Conv_Ovf_U4_Un;
public static final System.Reflection.Emit.OpCode Conv_Ovf_U8;
public static final System.Reflection.Emit.OpCode Conv_Ovf_U8_Un;
public static final System.Reflection.Emit.OpCode Conv_R_Un;
public static final System.Reflection.Emit.OpCode Conv_R4;
public static final System.Reflection.Emit.OpCode Conv_R8;
public static final System.Reflection.Emit.OpCode Conv_U;
public static final System.Reflection.Emit.OpCode Conv_U1;
public static final System.Reflection.Emit.OpCode Conv_U2;
public static final System.Reflection.Emit.OpCode Conv_U4;
public static final System.Reflection.Emit.OpCode Conv_U8;
public static final System.Reflection.Emit.OpCode Cpblk;
public static final System.Reflection.Emit.OpCode Cpobj;
public static final System.Reflection.Emit.OpCode Div;
public static final System.Reflection.Emit.OpCode Div_Un;
public static final System.Reflection.Emit.OpCode Dup;
public static final System.Reflection.Emit.OpCode Endfilter;
public static final System.Reflection.Emit.OpCode Endfinally;
public static final System.Reflection.Emit.OpCode Initblk;
public static final System.Reflection.Emit.OpCode Initobj;
public static final System.Reflection.Emit.OpCode Isinst;
public static final System.Reflection.Emit.OpCode Jmp;
public static final System.Reflection.Emit.OpCode Ldarg;
public static final System.Reflection.Emit.OpCode Ldarg_0;
public static final System.Reflection.Emit.OpCode Ldarg_1;
public static final System.Reflection.Emit.OpCode Ldarg_2;
public static final System.Reflection.Emit.OpCode Ldarg_3;
public static final System.Reflection.Emit.OpCode Ldarg_S;
public static final System.Reflection.Emit.OpCode Ldarga;
public static final System.Reflection.Emit.OpCode Ldarga_S;
public static final System.Reflection.Emit.OpCode Ldc_I4;
public static final System.Reflection.Emit.OpCode Ldc_I4_0;
public static final System.Reflection.Emit.OpCode Ldc_I4_1;
public static final System.Reflection.Emit.OpCode Ldc_I4_2;
public static final System.Reflection.Emit.OpCode Ldc_I4_3;
public static final System.Reflection.Emit.OpCode Ldc_I4_4;
public static final System.Reflection.Emit.OpCode Ldc_I4_5;
public static final System.Reflection.Emit.OpCode Ldc_I4_6;
public static final System.Reflection.Emit.OpCode Ldc_I4_7;
public static final System.Reflection.Emit.OpCode Ldc_I4_8;
public static final System.Reflection.Emit.OpCode Ldc_I4_M1;
public static final System.Reflection.Emit.OpCode Ldc_I4_S;
public static final System.Reflection.Emit.OpCode Ldc_I8;
public static final System.Reflection.Emit.OpCode Ldc_R4;
public static final System.Reflection.Emit.OpCode Ldc_R8;
public static final System.Reflection.Emit.OpCode Ldelem_I;
public static final System.Reflection.Emit.OpCode Ldelem_I1;
```

```
public static final System.Reflection.Emit.OpCode Ldelem_I2;
public static final System.Reflection.Emit.OpCode Ldelem_I4;
public static final System.Reflection.Emit.OpCode Ldelem_I8;
public static final System.Reflection.Emit.OpCode Ldelem_R4;
public static final System.Reflection.Emit.OpCode Ldelem_R8;
public static final System.Reflection.Emit.OpCode Ldelem_Ref;
public static final System.Reflection.Emit.OpCode Ldelem_U1;
public static final System.Reflection.Emit.OpCode Ldelem_U2;
public static final System.Reflection.Emit.OpCode Ldelem_U4;
public static final System.Reflection.Emit.OpCode Ldelema;
public static final System.Reflection.Emit.OpCode Ldfld;
public static final System.Reflection.Emit.OpCode Ldflda;
public static final System.Reflection.Emit.OpCode Ldftn;
public static final System.Reflection.Emit.OpCode Ldind_I;
public static final System.Reflection.Emit.OpCode Ldind_I1;
public static final System.Reflection.Emit.OpCode Ldind_I2;
public static final System.Reflection.Emit.OpCode Ldind_I4;
public static final System.Reflection.Emit.OpCode Ldind_I8;
public static final System.Reflection.Emit.OpCode Ldind_R4;
public static final System.Reflection.Emit.OpCode Ldind_R8;
public static final System.Reflection.Emit.OpCode Ldind_Ref;
public static final System.Reflection.Emit.OpCode Ldind_U1;
public static final System.Reflection.Emit.OpCode Ldind_U2;
public static final System.Reflection.Emit.OpCode Ldind_U4;
public static final System.Reflection.Emit.OpCode Ldlen;
public static final System.Reflection.Emit.OpCode Ldloc;
public static final System.Reflection.Emit.OpCode Ldloc_0;
public static final System.Reflection.Emit.OpCode Ldloc_1;
public static final System.Reflection.Emit.OpCode Ldloc_2;
public static final System.Reflection.Emit.OpCode Ldloc_3;
public static final System.Reflection.Emit.OpCode Ldloc_S;
public static final System.Reflection.Emit.OpCode Ldloca;
public static final System.Reflection.Emit.OpCode Ldloca_S;
public static final System.Reflection.Emit.OpCode Ldnull;
public static final System.Reflection.Emit.OpCode Ldobj;
public static final System.Reflection.Emit.OpCode Ldsfld;
public static final System.Reflection.Emit.OpCode Ldsflda;
public static final System.Reflection.Emit.OpCode Ldstr;
public static final System.Reflection.Emit.OpCode Ldtoken;
public static final System.Reflection.Emit.OpCode Ldvirtftn;
public static final System.Reflection.Emit.OpCode Leave;
public static final System.Reflection.Emit.OpCode Leave_S;
public static final System.Reflection.Emit.OpCode Localloc;
public static final System.Reflection.Emit.OpCode Mkrefany;
public static final System.Reflection.Emit.OpCode Mul;
public static final System.Reflection.Emit.OpCode Mul_Ovf;
public static final System.Reflection.Emit.OpCode Mul_Ovf_Un;
public static final System.Reflection.Emit.OpCode Neg;
public static final System.Reflection.Emit.OpCode Newarr;
public static final System.Reflection.Emit.OpCode Newobj;
public static final System.Reflection.Emit.OpCode Nop;
public static final System.Reflection.Emit.OpCode Not;
public static final System.Reflection.Emit.OpCode Or;
public static final System.Reflection.Emit.OpCode Pop;
public static final System.Reflection.Emit.OpCode Prefix1;
public static final System.Reflection.Emit.OpCode Prefix2;
public static final System.Reflection.Emit.OpCode Prefix3;
public static final System.Reflection.Emit.OpCode Prefix4;
public static final System.Reflection.Emit.OpCode Prefix5;
public static final System.Reflection.Emit.OpCode Prefix6;
public static final System.Reflection.Emit.OpCode Prefix7;
public static final System.Reflection.Emit.OpCode Prefixref;
public static final System.Reflection.Emit.OpCode Refanytype;
public static final System.Reflection.Emit.OpCode Refanyval;
public static final System.Reflection.Emit.OpCode Rem;
public static final System.Reflection.Emit.OpCode Rem_Un;
public static final System.Reflection.Emit.OpCode Ret;
public static final System.Reflection.Emit.OpCode Rethrow;
```

```
public static final System.Reflection.Emit.OpCode Shl;
public static final System.Reflection.Emit.OpCode Shr;
public static final System.Reflection.Emit.OpCode Shr_Un;
public static final System.Reflection.Emit.OpCode Sizeof;
public static final System.Reflection.Emit.OpCode Starg;
public static final System.Reflection.Emit.OpCode Starg_S;
public static final System.Reflection.Emit.OpCode Stelem_I;
public static final System.Reflection.Emit.OpCode Stelem_I1;
public static final System.Reflection.Emit.OpCode Stelem_I2;
public static final System.Reflection.Emit.OpCode Stelem_I4;
public static final System.Reflection.Emit.OpCode Stelem_I8;
public static final System.Reflection.Emit.OpCode Stelem_R4;
public static final System.Reflection.Emit.OpCode Stelem_R8;
public static final System.Reflection.Emit.OpCode Stelem_Ref;
public static final System.Reflection.Emit.OpCode Stfld;
public static final System.Reflection.Emit.OpCode Stind_I;
public static final System.Reflection.Emit.OpCode Stind_I1;
public static final System.Reflection.Emit.OpCode Stind_I2;
public static final System.Reflection.Emit.OpCode Stind_I4;
public static final System.Reflection.Emit.OpCode Stind_I8;
public static final System.Reflection.Emit.OpCode Stind_R4;
public static final System.Reflection.Emit.OpCode Stind_R8;
public static final System.Reflection.Emit.OpCode Stind_Ref;
public static final System.Reflection.Emit.OpCode Stloc;
public static final System.Reflection.Emit.OpCode Stloc_0;
public static final System.Reflection.Emit.OpCode Stloc_1;
public static final System.Reflection.Emit.OpCode Stloc_2;
public static final System.Reflection.Emit.OpCode Stloc_3;
public static final System.Reflection.Emit.OpCode Stloc_S;
public static final System.Reflection.Emit.OpCode Stobj;
public static final System.Reflection.Emit.OpCode Stsfld;
public static final System.Reflection.Emit.OpCode Sub;
public static final System.Reflection.Emit.OpCode Sub_Ovf;
public static final System.Reflection.Emit.OpCode Sub_Ovf_Un;
public static final System.Reflection.Emit.OpCode Switch;
public static final System.Reflection.Emit.OpCode Tailcall;
public static final System.Reflection.Emit.OpCode Throw;
public static final System.Reflection.Emit.OpCode Unaligned;
public static final System.Reflection.Emit.OpCode Unbox;
public static final System.Reflection.Emit.OpCode Volatile;
public static final System.Reflection.Emit.OpCode Xor;
```

OpCodeType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum OpCodeType
```

Members

Annotation

Macro

Nternal

Objmodel

Prefix

Primitive

OperandType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum OperandType
```

Members

InlineBrTarget

InlineField

InlineI

InlineI8

InlineMethod

InlineNone

InlinePhi

InlineR

InlineSig

InlineString

InlineSwitch

InlineTok

InlineType

InlineVar

ShortInlineBrTarget

ShortInlineI

ShortInlineR

ShortInlineVar

PackingSize Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PackingSize
```

Members

Size1

Size16

Size2

Size4

Size8

Unspecified

ParameterBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ParameterBuilder
   extends System.Object
```

Public Properties

```
public int get_Attributes();
public boolean get_IsIn();
public boolean get_IsOptional();
public boolean get_IsOut();
public System.String get_Name();
public int get_Position();
```

Public Methods

```
public System.Reflection.Emit.ParameterToken GetToken();
public void SetConstant(
   System.Object defaultValue
);
public void SetCustomAttribute(
   System.Reflection.Emit.CustomAttributeBuilder customBuilder
);
public void SetCustomAttribute(
   System.Reflection.ConstructorInfo con,
   ubyte[] binaryAttribute
);
public void SetMarshal(
   System.Reflection.Emit.UnmanagedMarshal unmanagedMarshal
);
```

ParameterToken Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ParameterToken
   extends System.ValueType
```

Public Properties

```
public int get_Token();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

Public Fields

```
public static final System.Reflection.Emit.ParameterToken Empty;
```

PEFileKinds Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PEFileKinds
```

Members

ConsoleApplication

Dll

WindowApplication

PropertyBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PropertyBuilder
   extends System.Reflection.PropertyInfo
```

Public Properties

```
public System.Reflection.PropertyAttributes get_Attributes();
public boolean get_CanRead();
public boolean get_CanWrite();
public System.Type get_DeclaringType();
public System.String get_Name();
public System.Reflection.Emit.PropertyToken get_PropertyToken();
public System.Type get_PropertyType();
public System.Type get_ReflectedType();
```

Public Methods

```
public void AddOtherMethod(
   System.Reflection.Emit.MethodBuilder mdBuilder
);
public System.Reflection.MethodInfo[] GetAccessors(
   boolean nonPublic
);
public System.Object[] GetCustomAttributes(
   boolean inherit
);
public System.Object[] GetCustomAttributes(
   System.Type attributeType,
   boolean inherit
);
public System.Reflection.MethodInfo GetGetMethod(
   boolean nonPublic
);
public System.Reflection.ParameterInfo[] GetIndexParameters();
public System.Reflection.MethodInfo GetSetMethod(
   boolean nonPublic
);
public System.Object GetValue(
   System.Object obj,
   System.Object[] index
);
public System.Object GetValue(
   System.Object obj,
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Object[] index,
   System.Globalization.CultureInfo culture
);
public boolean IsDefined(
   System.Type attributeType,
   boolean inherit
);
public void SetConstant(
   System.Object defaultValue
);
public void SetCustomAttribute(
```

```
      System.Reflection.Emit.CustomAttributeBuilder customBuilder
   );
   public void SetCustomAttribute(
      System.Reflection.ConstructorInfo con,
      ubyte[] binaryAttribute
   );
   public void SetGetMethod(
      System.Reflection.Emit.MethodBuilder mdBuilder
   );
   public void SetSetMethod(
      System.Reflection.Emit.MethodBuilder mdBuilder
   );
   public void SetValue(
      System.Object obj,
      System.Object value,
      System.Object[] index
   );
   public void SetValue(
      System.Object obj,
      System.Object value,
      System.Reflection.BindingFlags invokeAttr,
      System.Reflection.Binder binder,
      System.Object[] index,
      System.Globalization.CultureInfo culture
   );
```

PropertyToken Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class PropertyToken
   extends System.ValueType
```

Public Properties

```
public int get_Token();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

Public Fields

```
public static final System.Reflection.Emit.PropertyToken Empty;
```

SignatureHelper Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SignatureHelper
   extends System.Object
```

Public Methods

```
public void AddArgument(
   System.Type clsArgument
);
public void AddSentinel();
public boolean Equals(
   System.Object obj
);
public static System.Reflection.Emit.SignatureHelper GetFieldSigHelper(
   System.Reflection.Module mod
);
public int GetHashCode();
public static System.Reflection.Emit.SignatureHelper GetLocalVarSigHelper(
   System.Reflection.Module mod
);
public static System.Reflection.Emit.SignatureHelper GetMethodSigHelper(
   System.Reflection.Module mod,
   System.Reflection.CallingConventions callingConvention,
   System.Type returnType
);
public static System.Reflection.Emit.SignatureHelper GetMethodSigHelper(
   System.Reflection.Module mod,
   System.Runtime.InteropServices.CallingConvention unmanagedCallConv,
   System.Type returnType
);
public static System.Reflection.Emit.SignatureHelper GetMethodSigHelper(
   System.Reflection.Module mod,
   System.Type returnType,
   System.Type[] parameterTypes
);
public static System.Reflection.Emit.SignatureHelper GetPropertySigHelper(
   System.Reflection.Module mod,
   System.Type returnType,
   System.Type[] parameterTypes
);
public ubyte[] GetSignature();
public System.String ToString();
```

SignatureToken Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SignatureToken
   extends System.ValueType
```

Public Properties

```
public int get_Token();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

Public Fields

```
public static final System.Reflection.Emit.SignatureToken Empty;
```

StackBehaviour Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum StackBehaviour
```

Members

Pop0

Pop1

Pop1_pop1

Popi

Popi_pop1

Popi_popi

Popi_popi_popi

Popi_popi8

Popi_popr4

Popi_popr8

Popref

Popref_pop1

Popref_popi

Popref_popi_popi

Popref_popi_popi8

Popref_popi_popr4

Popref_popi_popr8

Popref_popi_popref

Push0

Push1

Push1_push1

Pushi

Pushi8

Pushr4

Pushr8

Pushref

Varpop

Varpush

StringToken Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StringToken
   extends System.ValueType
```

Public Properties

```
public int get_Token();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

TypeBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TypeBuilder
   extends System.Type
```

Public Properties

```
public System.Reflection.Assembly get_Assembly();
public System.String get_AssemblyQualifiedName();
public System.Type get_BaseType();
public System.Type get_DeclaringType();
public System.String get_FullName();
public System.Guid get_GUID();
public System.Reflection.Module get_Module();
public System.String get_Name();
public System.String get_Namespace();
public System.Reflection.Emit.PackingSize get_PackingSize();
public System.Type get_ReflectedType();
public int get_Size();
public System.RuntimeTypeHandle get_TypeHandle();
public System.Reflection.Emit.TypeToken get_TypeToken();
public System.Type get_UnderlyingSystemType();
```

Public Methods

```
public void AddDeclarativeSecurity(
   System.Security.Permissions.SecurityAction action,
   System.Security.PermissionSet pset
);
public void AddInterfaceImplementation(
   System.Type interfaceType
);
public System.Type CreateType();
public System.Reflection.Emit.ConstructorBuilder DefineConstructor(
   System.Reflection.MethodAttributes attributes,
   System.Reflection.CallingConventions callingConvention,
   System.Type[] parameterTypes
);
public System.Reflection.Emit.ConstructorBuilder DefineDefaultConstructor(
   System.Reflection.MethodAttributes attributes
);
public System.Reflection.Emit.EventBuilder DefineEvent(
   System.String name,
   System.Reflection.EventAttributes attributes,
   System.Type eventtype
);
public System.Reflection.Emit.FieldBuilder DefineField(
   System.String fieldName,
   System.Type type,
   System.Reflection.FieldAttributes attributes
);
public System.Reflection.Emit.FieldBuilder DefineInitializedData(
   System.String name,
   ubyte[] data,
   System.Reflection.FieldAttributes attributes
);
public System.Reflection.Emit.MethodBuilder DefineMethod(
```

```
   System.String name,
   System.Reflection.MethodAttributes attributes,
   System.Type returnType,
   System.Type[] parameterTypes
);
public System.Reflection.Emit.MethodBuilder DefineMethod(
   System.String name,
   System.Reflection.MethodAttributes attributes,
   System.Reflection.CallingConventions callingConvention,
   System.Type returnType,
   System.Type[] parameterTypes
);
public void DefineMethodOverride(
   System.Reflection.MethodInfo methodInfoBody,
   System.Reflection.MethodInfo methodInfoDeclaration
);
public System.Reflection.Emit.TypeBuilder DefineNestedType(
   System.String name
);
public System.Reflection.Emit.TypeBuilder DefineNestedType(
   System.String name,
   System.Reflection.TypeAttributes attr
);
public System.Reflection.Emit.TypeBuilder DefineNestedType(
   System.String name,
   System.Reflection.TypeAttributes attr,
   System.Type parent
);
public System.Reflection.Emit.TypeBuilder DefineNestedType(
   System.String name,
   System.Reflection.TypeAttributes attr,
   System.Type parent,
   int typeSize
);
public System.Reflection.Emit.TypeBuilder DefineNestedType(
   System.String name,
   System.Reflection.TypeAttributes attr,
   System.Type parent,
   System.Reflection.Emit.PackingSize packSize
);
public System.Reflection.Emit.TypeBuilder DefineNestedType(
   System.String name,
   System.Reflection.TypeAttributes attr,
   System.Type parent,
   System.Type[] interfaces
);
public System.Reflection.Emit.MethodBuilder DefinePInvokeMethod(
   System.String name,
   System.String dllName,
   System.Reflection.MethodAttributes attributes,
   System.Reflection.CallingConventions callingConvention,
   System.Type returnType,
   System.Type[] parameterTypes,
   System.Runtime.InteropServices.CallingConvention nativeCallConv,
   System.Runtime.InteropServices.CharSet nativeCharSet
);
public System.Reflection.Emit.MethodBuilder DefinePInvokeMethod(
   System.String name,
   System.String dllName,
   System.String entryName,
   System.Reflection.MethodAttributes attributes,
   System.Reflection.CallingConventions callingConvention,
   System.Type returnType,
   System.Type[] parameterTypes,
   System.Runtime.InteropServices.CallingConvention nativeCallConv,
   System.Runtime.InteropServices.CharSet nativeCharSet
);
public System.Reflection.Emit.PropertyBuilder DefineProperty(
```

```
   System.String name,
   System.Reflection.PropertyAttributes attributes,
   System.Type returnType,
   System.Type[] parameterTypes
);
public System.Reflection.Emit.ConstructorBuilder DefineTypeInitializer();
public System.Reflection.Emit.FieldBuilder DefineUninitializedData(
   System.String name,
   int size,
   System.Reflection.FieldAttributes attributes
);
public System.Reflection.ConstructorInfo[] GetConstructors(
   System.Reflection.BindingFlags bindingAttr
);
public System.Object[] GetCustomAttributes(
   boolean inherit
);
public System.Object[] GetCustomAttributes(
   System.Type attributeType,
   boolean inherit
);
public System.Type GetElementType();
public System.Reflection.EventInfo GetEvent(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.EventInfo[] GetEvents();
public System.Reflection.EventInfo[] GetEvents(
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.FieldInfo GetField(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.FieldInfo[] GetFields(
   System.Reflection.BindingFlags bindingAttr
);
public System.Type GetInterface(
   System.String name,
   boolean ignoreCase
);
public System.Reflection.InterfaceMapping GetInterfaceMap(
   System.Type interfaceType
);
public System.Type[] GetInterfaces();
public System.Reflection.MemberInfo[] GetMember(
   System.String name,
   System.Reflection.MemberTypes type,
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.MemberInfo[] GetMembers(
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.MethodInfo[] GetMethods(
   System.Reflection.BindingFlags bindingAttr
);
public System.Type GetNestedType(
   System.String name,
   System.Reflection.BindingFlags bindingAttr
);
public System.Type[] GetNestedTypes(
   System.Reflection.BindingFlags bindingAttr
);
public System.Reflection.PropertyInfo[] GetProperties(
   System.Reflection.BindingFlags bindingAttr
);
public System.Object InvokeMember(
   System.String name,
```

```
   System.Reflection.BindingFlags invokeAttr,
   System.Reflection.Binder binder,
   System.Object target,
   System.Object[] args,
   System.Reflection.ParameterModifier[] modifiers,
   System.Globalization.CultureInfo culture,
   System.String[] namedParameters
);
public boolean IsAssignableFrom(
   System.Type c
);
public boolean IsDefined(
   System.Type attributeType,
   boolean inherit
);
public boolean IsSubclassOf(
   System.Type c
);
public void SetCustomAttribute(
   System.Reflection.Emit.CustomAttributeBuilder customBuilder
);
public void SetCustomAttribute(
   System.Reflection.ConstructorInfo con,
   ubyte[] binaryAttribute
);
public void SetParent(
   System.Type parent
);
public System.String ToString();
```

Public Fields

```
public static int UnspecifiedTypeSize;
```

Protected Methods

```
protected System.Reflection.TypeAttributes GetAttributeFlagsImpl();
protected System.Reflection.ConstructorInfo GetConstructorImpl(
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Reflection.CallingConventions callConvention,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
protected System.Reflection.MethodInfo GetMethodImpl(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Reflection.CallingConventions callConvention,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
protected System.Reflection.PropertyInfo GetPropertyImpl(
   System.String name,
   System.Reflection.BindingFlags bindingAttr,
   System.Reflection.Binder binder,
   System.Type returnType,
   System.Type[] types,
   System.Reflection.ParameterModifier[] modifiers
);
protected boolean HasElementTypeImpl();
protected boolean IsArrayImpl();
protected boolean IsByRefImpl();
protected boolean IsCOMObjectImpl();
```

```
protected boolean IsPointerImpl();
protected boolean IsPrimitiveImpl();
```

TypeToken Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class TypeToken
   extends System.ValueType
```

Public Properties

```
public int get_Token();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

Public Fields

```
public static final System.Reflection.Emit.TypeToken Empty;
```

UnmanagedMarshal Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class UnmanagedMarshal
   extends System.Object
```

Public Properties

```
public System.Runtime.InteropServices.UnmanagedType get_BaseType();
public int get_ElementCount();
public System.Runtime.InteropServices.UnmanagedType get_GetUnmanagedType();
public System.Guid get_IIDGuid();
```

Public Methods

```
public static System.Reflection.Emit.UnmanagedMarshal DefineByValArray(
   int elemCount
);
public static System.Reflection.Emit.UnmanagedMarshal DefineByValTStr(
   int elemCount
);
public static System.Reflection.Emit.UnmanagedMarshal DefineLPArray(
   System.Runtime.InteropServices.UnmanagedType elemType
);
public static System.Reflection.Emit.UnmanagedMarshal DefineSafeArray(
   System.Runtime.InteropServices.UnmanagedType elemType
);
public static System.Reflection.Emit.UnmanagedMarshal DefineUnmanagedMarshal(
   System.Runtime.InteropServices.UnmanagedType unmanagedType
);
```

System.Resources Namespace (Visual J# Syntax)

Classes

MissingManifestResourceException

NeutralResourcesLanguageAttribute

ResourceManager

ResourceReader

ResourceSet

ResourceWriter

ResXFileRef

ResXFileRef.Converter

ResXResourceReader

ResXResourceSet

ResXResourceWriter

SatelliteContractVersionAttribute

Interfaces

IResourceReader

IResourceWriter

IResourceReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IResourceReader
   implements System.Collections.IEnumerable, System.IDisposable
```

Public Methods

```
void Close();
System.Collections.IDictionaryEnumerator GetEnumerator();
```

IResourceWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IResourceWriter
   implements System.IDisposable
```

Public Methods

```
void AddResource(
   System.String name,
   ubyte[] value
);
void AddResource(
   System.String name,
   System.Object value
);
void AddResource(
   System.String name,
   System.String value
);
void Close();
void Generate();
```

MissingManifestResourceException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class MissingManifestResourceException
   extends System.SystemException
```

Public Constructors

```
public MissingManifestResourceException();
public MissingManifestResourceException(
   System.String message
);
public MissingManifestResourceException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected MissingManifestResourceException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

NeutralResourcesLanguageAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class NeutralResourcesLanguageAttribute
   extends System.Attribute
```

Public Constructors

```
public NeutralResourcesLanguageAttribute(
   System.String cultureName
);
```

Public Properties

```
public System.String get_CultureName();
```

ResourceManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ResourceManager
   extends System.Object
```

Public Constructors

```
public ResourceManager(
   System.Type resourceSource
);
public ResourceManager(
   System.String baseName,
   System.Reflection.Assembly assembly
);
public ResourceManager(
   System.String baseName,
   System.Reflection.Assembly assembly,
   System.Type usingResourceSet
);
```

Public Properties

```
public System.String get_BaseName();
public boolean get_IgnoreCase();
public void set_IgnoreCase(
   boolean value
);
public System.Type get_ResourceSetType();
```

Public Methods

```
public static System.Resources.ResourceManager CreateFileBasedResourceManager(
   System.String baseName,
   System.String resourceDir,
   System.Type usingResourceSet
);
public System.Object GetObject(
   System.String name
);
public System.Object GetObject(
   System.String name,
   System.Globalization.CultureInfo culture
);
public System.Resources.ResourceSet GetResourceSet(
   System.Globalization.CultureInfo culture,
   boolean createIfNotExists,
   boolean tryParents
);
public System.String GetString(
   System.String name
);
public System.String GetString(
   System.String name,
   System.Globalization.CultureInfo culture
```

```
);
public void ReleaseAllResources();
```

Public Fields

```
public static final int HeaderVersionNumber;
public static final int MagicNumber;
```

Protected Constructors

```
protected ResourceManager();
```

Protected Methods

```
protected static System.Globalization.CultureInfo GetNeutralResourcesLanguage(
   System.Reflection.Assembly a
);
protected System.String GetResourceFileName(
   System.Globalization.CultureInfo culture
);
protected static System.Version GetSatelliteContractVersion(
   System.Reflection.Assembly a
);
protected System.Resources.ResourceSet InternalGetResourceSet(
   System.Globalization.CultureInfo culture,
   boolean createIfNotExists,
   boolean tryParents
);
```

Protected Fields

```
protected System.String BaseNameField;
protected System.Reflection.Assembly MainAssembly;
protected System.Collections.Hashtable ResourceSets;
```

ResourceReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ResourceReader
   extends System.Object
   implements System.Resources.IResourceReader, System.Collections.IEnumerable, System.IDispo
sable
```

Public Constructors

```
public ResourceReader(
   System.IO.Stream stream
);
public ResourceReader(
   System.String fileName
);
```

Public Methods

```
public final void Close();
public final System.Collections.IDictionaryEnumerator GetEnumerator();
```

ResourceSet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ResourceSet
   extends System.Object
   implements System.IDisposable
```

Public Constructors

```
public ResourceSet(
   System.IO.Stream stream
);
public ResourceSet(
   System.Resources.IResourceReader reader
);
public ResourceSet(
   System.String fileName
);
```

Public Methods

```
public void Close();
public final void Dispose();
public System.Type GetDefaultReader();
public System.Type GetDefaultWriter();
public System.Object GetObject(
   System.String name
);
public System.Object GetObject(
   System.String name,
   boolean ignoreCase
);
public System.String GetString(
   System.String name
);
public System.String GetString(
   System.String name,
   boolean ignoreCase
);
```

Protected Constructors

```
protected ResourceSet();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void ReadResources();
```

Protected Fields

```
protected System.Resources.IResourceReader Reader;
protected System.Collections.Hashtable Table;
```

ResourceWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ResourceWriter
   extends System.Object
   implements System.Resources.IResourceWriter, System.IDisposable
```

Public Constructors

```
public ResourceWriter(
   System.IO.Stream stream
);
public ResourceWriter(
   System.String fileName
);
```

Public Methods

```
public final void AddResource(
   System.String name,
   ubyte[] value
);
public final void AddResource(
   System.String name,
   System.Object value
);
public final void AddResource(
   System.String name,
   System.String value
);
public final void Close();
public final void Dispose();
public final void Generate();
```

ResXFileRef Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Resou
rces.ResXFileRef+Converter, System.Windows.Forms, Version=1.0.3300.0, Culture=neutral, Public
KeyToken=b77a5c561934e089") */
public class ResXFileRef
   extends System.Object
```

Public Constructors

```
public ResXFileRef(
   System.String fileName,
   System.String typeName
);
```

Public Methods

```
public System.String ToString();
```

ResXFileRef.Converter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Converter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public Converter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

ResXResourceReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ResXResourceReader
   extends System.Object
   implements System.Resources.IResourceReader, System.Collections.IEnumerable, System.IDispo
sable
```

Public Constructors

```
public ResXResourceReader(
   System.IO.Stream stream
);
public ResXResourceReader(
   System.IO.TextReader reader
);
public ResXResourceReader(
   System.String fileName
);
public ResXResourceReader(
   System.IO.Stream stream,
   System.ComponentModel.Design.ITypeResolutionService typeResolver
);
public ResXResourceReader(
   System.IO.TextReader reader,
   System.ComponentModel.Design.ITypeResolutionService typeResolver
);
public ResXResourceReader(
   System.String fileName,
   System.ComponentModel.Design.ITypeResolutionService typeResolver
);
```

Public Methods

```
public final void Close();
public static System.Resources.ResXResourceReader FromFileContents(
   System.String fileContents
);
public static System.Resources.ResXResourceReader FromFileContents(
   System.String fileContents,
   System.ComponentModel.Design.ITypeResolutionService typeResolver
);
public final System.Collections.IDictionaryEnumerator GetEnumerator();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

ResXResourceSet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ResXResourceSet
   extends System.Resources.ResourceSet
```

Public Constructors

```
public ResXResourceSet(
   System.IO.Stream stream
);
public ResXResourceSet(
   System.String fileName
);
```

Public Methods

```
public System.Type GetDefaultReader();
public System.Type GetDefaultWriter();
```

ResXResourceWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ResXResourceWriter
   extends System.Object
   implements System.Resources.IResourceWriter, System.IDisposable
```

Public Constructors

```
public ResXResourceWriter(
   System.IO.Stream stream
);
public ResXResourceWriter(
   System.IO.TextWriter textWriter
);
public ResXResourceWriter(
   System.String fileName
);
```

Public Methods

```
public final void AddResource(
   System.String name,
   ubyte[] value
);
public final void AddResource(
   System.String name,
   System.Object value
);
public final void AddResource(
   System.String name,
   System.String value
);
public final void Close();
public void Dispose();
public final void Generate();
```

Public Fields

```
public static final System.String BinSerializedObjectMimeType;
public static final System.String ByteArraySerializedObjectMimeType;
public static final System.String DefaultSerializedObjectMimeType;
public static final System.String ResMimeType;
public static final System.String ResourceSchema;
public static final System.String SoapSerializedObjectMimeType;
public static final System.String Version;
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

SatelliteContractVersionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = true) */
public class SatelliteContractVersionAttribute
   extends System.Attribute
```

Public Constructors

```
public SatelliteContractVersionAttribute(
   System.String version
);
```

Public Properties

```
public System.String get_Version();
```

System.Runtime.CompilerServices Namespace (Visual J# Syntax)

Classes

AccessedThroughPropertyAttribute

CallConvCdecl

CallConvFastcall

CallConvStdcall

CallConvThiscall

CompilationRelaxationsAttribute

CompilerGlobalScopeAttribute

CustomConstantAttribute

DateTimeConstantAttribute

DecimalConstantAttribute

DiscardableAttribute

IDispatchConstantAttribute

IndexerNameAttribute

IsVolatile

IUnknownConstantAttribute

MethodImplAttribute

RequiredAttributeAttribute

RuntimeHelpers

Enumerations

MethodCodeType

MethodImplOptions

AccessedThroughPropertyAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, AllowMultiple = false, Inherit
ed = true) */
public class AccessedThroughPropertyAttribute
   extends System.Attribute
```

Public Constructors

```
public AccessedThroughPropertyAttribute(
   System.String propertyName
);
```

Public Properties

```
public System.String get_PropertyName();
```

CallConvCdecl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CallConvCdecl
   extends System.Object
```

Public Constructors

```
public CallConvCdecl();
```

CallConvFastcall Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CallConvFastcall
   extends System.Object
```

Public Constructors

```
public CallConvFastcall();
```

CallConvStdcall Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CallConvStdcall
   extends System.Object
```

Public Constructors

```
public CallConvStdcall();
```

CallConvThiscall Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CallConvThiscall
   extends System.Object
```

Public Constructors

```
public CallConvThiscall();
```

CompilationRelaxationsAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Module, AllowMultiple = false, Inheri
ted = true) */
public class CompilationRelaxationsAttribute
   extends System.Attribute
```

Public Constructors

```
public CompilationRelaxationsAttribute(
   int relaxations
);
```

Public Properties

```
public int get_CompilationRelaxations();
```

CompilerGlobalScopeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class CompilerGlobalScopeAttribute
   extends System.Attribute
```

Public Constructors

```
public CompilerGlobalScopeAttribute();
```

CustomConstantAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, Parameter, AllowMultiple = fal
se, Inherited = false) */
public abstract class CustomConstantAttribute
   extends System.Attribute
```

Public Properties

```
public abstract System.Object get_Value();
```

Protected Constructors

```
protected CustomConstantAttribute();
```

DateTimeConstantAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, Parameter, AllowMultiple = fal
se, Inherited = false) */
public class DateTimeConstantAttribute
   extends System.Runtime.CompilerServices.CustomConstantAttribute
```

Public Constructors

```
public DateTimeConstantAttribute(
   long ticks
);
```

Public Properties

```
public System.Object get_Value();
```

DecimalConstantAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, Parameter, AllowMultiple = fal
se, Inherited = false) */
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public class DecimalConstantAttribute
   extends System.Attribute
```

Public Constructors

```
public DecimalConstantAttribute(
   ubyte scale,
   ubyte sign,
   System.UInt32 hi,
   System.UInt32 mid,
   System.UInt32 low
);
```

Public Properties

```
public System.Decimal get_Value();
```

DiscardableAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DiscardableAttribute
   extends System.Attribute
```

Public Constructors

```
public DiscardableAttribute();
```

IDispatchConstantAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, Parameter, AllowMultiple = fal
se, Inherited = false) */
public class IDispatchConstantAttribute
   extends System.Runtime.CompilerServices.CustomConstantAttribute
```

Public Constructors

```
public IDispatchConstantAttribute();
```

Public Properties

```
public System.Object get_Value();
```

IndexerNameAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, AllowMultiple = false, Inhe
rited = true) */
public class IndexerNameAttribute
   extends System.Attribute
```

Public Constructors

```
public IndexerNameAttribute(
   System.String indexerName
);
```

IsVolatile Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class IsVolatile
   extends System.Object
```

IUnknownConstantAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, Parameter, AllowMultiple = fal
se, Inherited = false) */
public class IUnknownConstantAttribute
   extends System.Runtime.CompilerServices.CustomConstantAttribute
```

Public Constructors

```
public IUnknownConstantAttribute();
```

Public Properties

```
public System.Object get_Value();
```

MethodCodeType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MethodCodeType
```

Members

IL

Native

OPTIL

Runtime

MethodImplAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Constructor, Method, AllowMultiple =
false, Inherited = false) */
public class MethodImplAttribute
   extends System.Attribute
```

Public Constructors

```
public MethodImplAttribute();
public MethodImplAttribute(
   short value
);
public MethodImplAttribute(
   System.Runtime.CompilerServices.MethodImplOptions methodImplOptions
);
```

Public Properties

```
public System.Runtime.CompilerServices.MethodImplOptions get_Value();
```

Public Fields

```
public System.Runtime.CompilerServices.MethodCodeType MethodCodeType;
```

MethodImplOptions Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MethodImplOptions
```

Members

ForwardRef

InternalCall

NoInlining

PreserveSig

Synchronized

Unmanaged

RequiredAttributeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, Enum, Interface, Allow
Multiple = false, Inherited = false) */
public class RequiredAttributeAttribute
   extends System.Attribute
```

Public Constructors

```
public RequiredAttributeAttribute(
   System.Type requiredContract
);
```

Public Properties

```
public System.Type get_RequiredContract();
```

RuntimeHelpers Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RuntimeHelpers
   extends System.Object
```

Public Properties

```
public static int get_OffsetToStringData();
```

Public Methods

```
public static System.Object GetObjectValue(
   System.Object obj
);
public static void InitializeArray(
   System.Array array,
   System.RuntimeFieldHandle fldHandle
);
public static void RunClassConstructor(
   System.RuntimeTypeHandle type
);
```

System.Runtime.InteropServices Namespace (Visual J# Syntax)

Classes

AutomationProxyAttribute

ClassInterfaceAttribute

CoClassAttribute

ComAliasNameAttribute

ComConversionLossAttribute

ComEventInterfaceAttribute

COMException

ComImportAttribute

ComRegisterFunctionAttribute

ComSourceInterfacesAttribute

ComUnregisterFunctionAttribute

ComVisibleAttribute

CurrencyWrapper

DispatchWrapper

DispIdAttribute

DllImportAttribute

ErrorWrapper

ExtensibleClassFactory

ExternalException

FieldOffsetAttribute

GuidAttribute

IDispatchImplAttribute

ImportedFromTypeLibAttribute

InAttribute

InterfaceTypeAttribute

InvalidComObjectException

InvalidOleVariantTypeException

LCIDConversionAttribute

Marshal

MarshalAsAttribute

MarshalDirectiveException

OptionalAttribute

OutAttribute

PreserveSigAttribute

PrimaryInteropAssemblyAttribute

ProgIdAttribute

RegistrationServices

RuntimeEnvironment

SafeArrayRankMismatchException

SafeArrayTypeMismatchException

SEHException

StructLayoutAttribute

TypeLibConverter

TypeLibFuncAttribute

TypeLibTypeAttribute

TypeLibVarAttribute

UnknownWrapper

Interfaces

ICustomAdapter

ICustomFactory

ICustomMarshaler

IRegistrationServices

ITypeLibConverter

ITypeLibExporterNameProvider

ITypeLibExporterNotifySink

ITypeLibImporterNotifySink

UCOMIBindCtx

UCOMIConnectionPoint

UCOMIConnectionPointContainer

UCOMIEnumConnectionPoints

UCOMIEnumConnections

UCOMIEnumMoniker

UCOMIEnumString

UCOMIEnumVARIANT

UCOMIMoniker

UCOMIPersistFile

UCOMIRunningObjectTable

UCOMIStream

UCOMITypeComp

UCOMITypeInfo

UCOMITypeLib

Structures

ArrayWithOffset

BIND_OPTS

BINDPTR

CONNECTDATA

DISPPARAMS

ELEMDESC

ELEMDESC.DESCUNION

EXCEPINFO

FILETIME

FUNCDESC

GCHandle

HandleRef

IDLDESC

PARAMDESC

STATSTG

TYPEATTR

TYPEDESC

TYPELIBATTR

VARDESC

VARDESC.DESCUNION

Delegates

ObjectCreationDelegate

Enumerations

AssemblyRegistrationFlags

CALLCONV

CallingConvention

CharSet

ClassInterfaceType

ComInterfaceType

ComMemberType

DESCKIND

ExporterEventKind

FUNCFLAGS

FUNCKIND

GCHandleType

IDispatchImplType

IDLFLAG

IMPLTYPEFLAGS

ImporterEventKind

INVOKEKIND

LayoutKind

LIBFLAGS

PARAMFLAG

SYSKIND

TYPEFLAGS

TYPEKIND

TypeLibExporterFlags

TypeLibFuncFlags

TypeLibImporterFlags

TypeLibTypeFlags

TypeLibVarFlags

UnmanagedType

VarEnum

VARFLAGS

ArrayWithOffset Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ArrayWithOffset
   extends System.ValueType
```

Public Constructors

```
public ArrayWithOffset(
   System.Object array,
   int offset
);
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public System.Object GetArray();
public int GetHashCode();
public int GetOffset();
```

AssemblyRegistrationFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum AssemblyRegistrationFlags
```

Members

None

SetCodeBase

AutomationProxyAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Interface, AllowMult
iple = false, Inherited = false) */
public class AutomationProxyAttribute
   extends System.Attribute
```

Public Constructors

```
public AutomationProxyAttribute(
   boolean val
);
```

Public Properties

```
public boolean get_Value();
```

BIND_OPTS Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class BIND_OPTS
   extends System.ValueType
```

Public Fields

```
public int cbStruct;
public int dwTickCountDeadline;
public int grfFlags;
public int grfMode;
```

BINDPTR Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class BINDPTR
   extends System.ValueType
```

Public Fields

```
public System.IntPtr lpfuncdesc;
public System.IntPtr lptcomp;
public System.IntPtr lpvardesc;
```

CALLCONV Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public enum CALLCONV
```

Members

CC_CDECL

CC_MACPASCAL

CC_MAX

CC_MPWCDECL

CC_MPWPASCAL

CC_MSCPASCAL

CC_PASCAL

CC_RESERVED

CC_STDCALL

CC_SYSCALL

CallingConvention Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CallingConvention
```

Members

Cdecl

FastCall

StdCall

ThisCall

Winapi

CharSet Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CharSet
```

Members

Ansi

Auto

None

Unicode

ClassInterfaceAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, AllowMultiple = fals
e, Inherited = false) */
public class ClassInterfaceAttribute
   extends System.Attribute
```

Public Constructors

```
public ClassInterfaceAttribute(
   short classInterfaceType
);
public ClassInterfaceAttribute(
   System.Runtime.InteropServices.ClassInterfaceType classInterfaceType
);
```

Public Properties

```
public System.Runtime.InteropServices.ClassInterfaceType get_Value();
```

ClassInterfaceType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ClassInterfaceType
```

Members

AutoDispatch

AutoDual

None

CoClassAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Interface, AllowMultiple = false, Inh
erited = false) */
public class CoClassAttribute
   extends System.Attribute
```

Public Constructors

```
public CoClassAttribute(
   System.Type coClass
);
```

Public Properties

```
public System.Type get_CoClass();
```

ComAliasNameAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Parameter, ReturnVal
ue, AllowMultiple = false, Inherited = false) */
public class ComAliasNameAttribute
   extends System.Attribute
```

Public Constructors

```
public ComAliasNameAttribute(
   System.String alias
);
```

Public Properties

```
public System.String get_Value();
```

ComConversionLossAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= false) */
public class ComConversionLossAttribute
   extends System.Attribute
```

Public Constructors

```
public ComConversionLossAttribute();
```

ComEventInterfaceAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Interface, AllowMultiple = false, Inh
erited = false) */
public class ComEventInterfaceAttribute
   extends System.Attribute
```

Public Constructors

```
public ComEventInterfaceAttribute(
   System.Type SourceInterface,
   System.Type EventProvider
);
```

Public Properties

```
public System.Type get_EventProvider();
public System.Type get_SourceInterface();
```

COMException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class COMException
   extends System.Runtime.InteropServices.ExternalException
```

Public Constructors

```
public COMException();
public COMException(
   System.String message
);
public COMException(
   System.String message,
   int errorCode
);
public COMException(
   System.String message,
   System.Exception inner
);
```

Public Methods

```
public System.String ToString();
```

Protected Constructors

```
protected COMException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ComImportAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Interface, AllowMultiple = fal
se, Inherited = false) */
public class ComImportAttribute
   extends System.Attribute
```

Public Constructors

```
public ComImportAttribute();
```

ComInterfaceType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ComInterfaceType
```

Members

InterfaceIsDual

InterfaceIsIDispatch

InterfaceIsIUnknown

ComMemberType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ComMemberType
```

Members

Method

PropGet

PropSet

ComRegisterFunctionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = false) */
public class ComRegisterFunctionAttribute
   extends System.Attribute
```

Public Constructors

```
public ComRegisterFunctionAttribute();
```

ComSourceInterfacesAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = false) */
public class ComSourceInterfacesAttribute
   extends System.Attribute
```

Public Constructors

```
public ComSourceInterfacesAttribute(
   System.String sourceInterfaces
);
public ComSourceInterfacesAttribute(
   System.Type sourceInterface
);
public ComSourceInterfacesAttribute(
   System.Type sourceInterface1,
   System.Type sourceInterface2
);
public ComSourceInterfacesAttribute(
   System.Type sourceInterface1,
   System.Type sourceInterface2,
   System.Type sourceInterface3
);
public ComSourceInterfacesAttribute(
   System.Type sourceInterface1,
   System.Type sourceInterface2,
   System.Type sourceInterface3,
   System.Type sourceInterface4
);
```

Public Properties

```
public System.String get_Value();
```

ComUnregisterFunctionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = false) */
public class ComUnregisterFunctionAttribute
   extends System.Attribute
```

Public Constructors

```
public ComUnregisterFunctionAttribute();
```

ComVisibleAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Enum, Method
, Property, Field, Interface, Delegate, AllowMultiple = false, Inherited = false) */
public class ComVisibleAttribute
   extends System.Attribute
```

Public Constructors

```
public ComVisibleAttribute(
   boolean visibility
);
```

Public Properties

```
public boolean get_Value();
```

CONNECTDATA Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class CONNECTDATA
   extends System.ValueType
```

Public Fields

```
public int dwCookie;
public System.Object pUnk;
```

CurrencyWrapper Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CurrencyWrapper
   extends System.Object
```

Public Constructors

```
public CurrencyWrapper(
   System.Decimal obj
);
public CurrencyWrapper(
   System.Object obj
);
```

Public Properties

```
public System.Decimal get_WrappedObject();
```

DESCKIND Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public enum DESCKIND
```

Members

DESCKIND_FUNCDESC

DESCKIND_IMPLICITAPPOBJ

DESCKIND_MAX

DESCKIND_NONE

DESCKIND_TYPECOMP

DESCKIND_VARDESC

DispatchWrapper Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DispatchWrapper
   extends System.Object
```

Public Constructors

```
public DispatchWrapper(
   System.Object obj
);
```

Public Properties

```
public System.Object get_WrappedObject();
```

DispIdAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, Property, Field, Event, Allow
Multiple = false, Inherited = false) */
public class DispIdAttribute
   extends System.Attribute
```

Public Constructors

```
public DispIdAttribute(
   int dispId
);
```

Public Properties

```
public int get_Value();
```

DISPPARAMS Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class DISPPARAMS
   extends System.ValueType
```

Public Fields

```
public int cArgs;
public int cNamedArgs;
public System.IntPtr rgdispidNamedArgs;
public System.IntPtr rgvarg;
```

DllImportAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = false) */
public class DllImportAttribute
   extends System.Attribute
```

Public Constructors

```
public DllImportAttribute(
   System.String dllName
);
```

Public Properties

```
public System.String get_Value();
```

Public Fields

```
public System.Runtime.InteropServices.CallingConvention CallingConvention;
public System.Runtime.InteropServices.CharSet CharSet;
public System.String EntryPoint;
public boolean ExactSpelling;
public boolean PreserveSig;
public boolean SetLastError;
```

ELEMDESC Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class ELEMDESC
   extends System.ValueType
```

Public Fields

```
public System.Runtime.InteropServices.ELEMDESC+DESCUNION desc;
public System.Runtime.InteropServices.TYPEDESC tdesc;
```

ELEMDESC.DESCUNION Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class DESCUNION
   extends System.ValueType
```

Public Fields

```
public System.Runtime.InteropServices.IDLDESC idldesc;
public System.Runtime.InteropServices.PARAMDESC paramdesc;
```

ErrorWrapper Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ErrorWrapper
   extends System.Object
```

Public Constructors

```
public ErrorWrapper(
   int errorCode
);
public ErrorWrapper(
   System.Exception e
);
public ErrorWrapper(
   System.Object errorCode
);
```

Public Properties

```
public int get_ErrorCode();
```

EXCEPINFO Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class EXCEPINFO
   extends System.ValueType
```

Public Fields

```
public System.String bstrDescription;
public System.String bstrHelpFile;
public System.String bstrSource;
public int dwHelpContext;
public System.IntPtr pfnDeferredFillIn;
public System.IntPtr pvReserved;
public short wCode;
public short wReserved;
```

ExporterEventKind Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ExporterEventKind
```

Members

ERROR_REFTOINVALIDASSEMBLY

NOTIF_CONVERTWARNING

NOTIF_TYPECONVERTED

ExtensibleClassFactory Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ExtensibleClassFactory
   extends System.Object
```

Public Methods

```
public static void RegisterObjectCreationCallback(
   System.Runtime.InteropServices.ObjectCreationDelegate callback
);
```

ExternalException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ExternalException
   extends System.SystemException
```

Public Constructors

```
public ExternalException();
public ExternalException(
   System.String message
);
public ExternalException(
   System.String message,
   int errorCode
);
public ExternalException(
   System.String message,
   System.Exception inner
);
```

Public Properties

```
public int get_ErrorCode();
```

Protected Constructors

```
protected ExternalException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

FieldOffsetAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, AllowMultiple = false, Inherit
ed = false) */
public class FieldOffsetAttribute
   extends System.Attribute
```

Public Constructors

```
public FieldOffsetAttribute(
   int offset
);
```

Public Properties

```
public int get_Value();
```

FILETIME Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class FILETIME
   extends System.ValueType
```

Public Fields

```
public int dwHighDateTime;
public int dwLowDateTime;
```

FUNCDESC Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class FUNCDESC
   extends System.ValueType
```

Public Fields

```
public System.Runtime.InteropServices.CALLCONV callconv;
public short cParams;
public short cParamsOpt;
public short cScodes;
public System.Runtime.InteropServices.ELEMDESC elemdescFunc;
public System.Runtime.InteropServices.FUNCKIND funckind;
public System.Runtime.InteropServices.INVOKEKIND invkind;
public System.IntPtr lprgelemdescParam;
public System.IntPtr lprgscode;
public int memid;
public short oVft;
public short wFuncFlags;
```

FUNCFLAGS Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
/** @attribute System.FlagsAttribute() */
public enum FUNCFLAGS
```

Members

FUNCFLAG_FBINDABLE

FUNCFLAG_FDEFAULTBIND

FUNCFLAG_FDEFAULTCOLLELEM

FUNCFLAG_FDISPLAYBIND

FUNCFLAG_FHIDDEN

FUNCFLAG_FIMMEDIATEBIND

FUNCFLAG_FNONBROWSABLE

FUNCFLAG_FREPLACEABLE

FUNCFLAG_FREQUESTEDIT

FUNCFLAG_FRESTRICTED

FUNCFLAG_FSOURCE

FUNCFLAG_FUIDEFAULT

FUNCFLAG_FUSESGETLASTERROR

FUNCKIND Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public enum FUNCKIND
```

Members

FUNC_DISPATCH

FUNC_NONVIRTUAL

FUNC_PUREVIRTUAL

FUNC_STATIC

FUNC_VIRTUAL

GCHandle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class GCHandle
   extends System.ValueType
```

Public Properties

```
public boolean get_IsAllocated();
public System.Object get_Target();
public void set_Target(
   System.Object value
);
```

Public Methods

```
public System.IntPtr AddrOfPinnedObject();
public static System.Runtime.InteropServices.GCHandle Alloc(
   System.Object value
);
public static System.Runtime.InteropServices.GCHandle Alloc(
   System.Object value,
   System.Runtime.InteropServices.GCHandleType type
);
public void Free();
```

Public Operators

```
public static System.IntPtr op_Explicit(
   System.Runtime.InteropServices.GCHandle value
);
public static System.Runtime.InteropServices.GCHandle op_Explicit(
   System.IntPtr value
);
```

GCHandleType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum GCHandleType
```

Members

Normal

Pinned

Weak

WeakTrackResurrection

GuidAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Enum, Interf
ace, Delegate, AllowMultiple = false, Inherited = false) */
public class GuidAttribute
   extends System.Attribute
```

Public Constructors

```
public GuidAttribute(
   System.String guid
);
```

Public Properties

```
public System.String get_Value();
```

HandleRef Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HandleRef
   extends System.ValueType
```

Public Constructors

```
public HandleRef(
   System.Object wrapper,
   System.IntPtr handle
);
```

Public Properties

```
public System.IntPtr get_Handle();
public System.Object get_Wrapper();
```

Public Operators

```
public static System.IntPtr op_Explicit(
   System.Runtime.InteropServices.HandleRef value
);
```

ICustomAdapter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICustomAdapter
```

Public Methods

```
System.Object GetUnderlyingObject();
```

ICustomFactory Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICustomFactory
```

Public Methods

```
System.MarshalByRefObject CreateInstance(
   System.Type serverType
);
```

ICustomMarshaler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICustomMarshaler
```

Public Methods

```
void CleanUpManagedData(
   System.Object ManagedObj
);
void CleanUpNativeData(
   System.IntPtr pNativeData
);
int GetNativeDataSize();
System.IntPtr MarshalManagedToNative(
   System.Object ManagedObj
);
System.Object MarshalNativeToManaged(
   System.IntPtr pNativeData
);
```

IDispatchImplAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, AllowMultiple = fals
e, Inherited = false) */
public class IDispatchImplAttribute
   extends System.Attribute
```

Public Constructors

```
public IDispatchImplAttribute(
   short implType
);
public IDispatchImplAttribute(
   System.Runtime.InteropServices.IDispatchImplType implType
);
```

Public Properties

```
public System.Runtime.InteropServices.IDispatchImplType get_Value();
```

IDispatchImplType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum IDispatchImplType
```

Members

CompatibleImpl

InternalImpl

SystemDefinedImpl

IDLDESC Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class IDLDESC
   extends System.ValueType
```

Public Fields

```
public int dwReserved;
public System.Runtime.InteropServices.IDLFLAG wIDLFlags;
```

IDLFLAG Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
/** @attribute System.FlagsAttribute() */
public enum IDLFLAG
```

Members

IDLFLAG_FIN

IDLFLAG_FLCID

IDLFLAG_FOUT

IDLFLAG_FRETVAL

IDLFLAG_NONE

IMPLTYPEFLAGS Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public enum IMPLTYPEFLAGS
```

Members

IMPLTYPEFLAG_FDEFAULT

IMPLTYPEFLAG_FDEFAULTVTABLE

IMPLTYPEFLAG_FRESTRICTED

IMPLTYPEFLAG_FSOURCE

ImportedFromTypeLibAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = false) */
public class ImportedFromTypeLibAttribute
   extends System.Attribute
```

Public Constructors

```
public ImportedFromTypeLibAttribute(
   System.String tlbFile
);
```

Public Properties

```
public System.String get_Value();
```

ImporterEventKind Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ImporterEventKind
```

Members

ERROR_REFTOINVALIDTYPELIB

NOTIF_CONVERTWARNING

NOTIF_TYPECONVERTED

InAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Parameter, AllowMultiple = false, Inh
erited = false) */
public class InAttribute
   extends System.Attribute
```

Public Constructors

```
public InAttribute();
```

InterfaceTypeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Interface, AllowMultiple = false, Inh
erited = false) */
public class InterfaceTypeAttribute
   extends System.Attribute
```

Public Constructors

```
public InterfaceTypeAttribute(
   short interfaceType
);
public InterfaceTypeAttribute(
   System.Runtime.InteropServices.ComInterfaceType interfaceType
);
```

Public Properties

```
public System.Runtime.InteropServices.ComInterfaceType get_Value();
```

InvalidComObjectException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class InvalidComObjectException
   extends System.SystemException
```

Public Constructors

```
public InvalidComObjectException();
public InvalidComObjectException(
   System.String message
);
public InvalidComObjectException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected InvalidComObjectException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

InvalidOleVariantTypeException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class InvalidOleVariantTypeException
   extends System.SystemException
```

Public Constructors

```
public InvalidOleVariantTypeException();
public InvalidOleVariantTypeException(
   System.String message
);
public InvalidOleVariantTypeException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected InvalidOleVariantTypeException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

INVOKEKIND Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public enum INVOKEKIND
```

Members

INVOKE_FUNC

INVOKE_PROPERTYGET

INVOKE_PROPERTYPUT

INVOKE_PROPERTYPUTREF

IRegistrationServices Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "CCBD682C-73A5-4568-B8B0-
C7007E11ABA2") */
public interface IRegistrationServices
```

Public Methods

```
System.Guid GetManagedCategoryGuid();
System.String GetProgIdForType(
   System.Type type
);
System.Type[] GetRegistrableTypesInAssembly(
   System.Reflection.Assembly assembly
);
boolean RegisterAssembly(
   System.Reflection.Assembly assembly,
   System.Runtime.InteropServices.AssemblyRegistrationFlags flags
);
void RegisterTypeForComClients(
   System.Type type,
   System.Guid& g
);
boolean TypeRepresentsComType(
   System.Type type
);
boolean TypeRequiresRegistration(
   System.Type type
);
boolean UnregisterAssembly(
   System.Reflection.Assembly assembly
);
```

ITypeLibConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "F1C3BF78-C3E4-11d3-88E7-
00902754C43A") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface ITypeLibConverter
```

Public Methods

```
System.Object ConvertAssemblyToTypeLib(
   System.Reflection.Assembly assembly,
   System.String typeLibName,
   System.Runtime.InteropServices.TypeLibExporterFlags flags,
   System.Runtime.InteropServices.ITypeLibExporterNotifySink notifySink
);
System.Reflection.Emit.AssemblyBuilder ConvertTypeLibToAssembly(
   System.Object typeLib,
   System.String asmFileName,
   int flags,
   System.Runtime.InteropServices.ITypeLibImporterNotifySink notifySink,
   ubyte[] publicKey,
   System.Reflection.StrongNameKeyPair keyPair,
   boolean unsafeInterfaces
);
System.Reflection.Emit.AssemblyBuilder ConvertTypeLibToAssembly(
   System.Object typeLib,
   System.String asmFileName,
   System.Runtime.InteropServices.TypeLibImporterFlags flags,
   System.Runtime.InteropServices.ITypeLibImporterNotifySink notifySink,
   ubyte[] publicKey,
   System.Reflection.StrongNameKeyPair keyPair,
   System.String asmNamespace,
   System.Version asmVersion
);
boolean GetPrimaryInteropAssembly(
   System.Guid g,
   int major,
   int minor,
   int lcid,
   System.String& asmName,
   System.String& asmCodeBase
);
```

ITypeLibExporterNameProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "FA1F3615-ACB9-486d-9EAC-
1BEF87E36B09") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface ITypeLibExporterNameProvider
```

Public Methods

```
System.String[] GetNames();
```

ITypeLibExporterNotifySink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "F1C3BF77-C3E4-11d3-88E7-
00902754C43A") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface ITypeLibExporterNotifySink
```

Public Methods

```
void ReportEvent(
   System.Runtime.InteropServices.ExporterEventKind eventKind,
   int eventCode,
   System.String eventMsg
);
System.Object ResolveRef(
   System.Reflection.Assembly assembly
);
```

ITypeLibImporterNotifySink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "F1C3BF76-C3E4-11d3-88E7-
00902754C43A") */
public interface ITypeLibImporterNotifySink
```

Public Methods

```
void ReportEvent(
   System.Runtime.InteropServices.ImporterEventKind eventKind,
   int eventCode,
   System.String eventMsg
);
System.Reflection.Assembly ResolveRef(
   System.Object typeLib
);
```

LayoutKind Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum LayoutKind
```

Members

Auto

Explicit

Sequential

LCIDConversionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = false) */
public class LCIDConversionAttribute
   extends System.Attribute
```

Public Constructors

```
public LCIDConversionAttribute(
   int lcid
);
```

Public Properties

```
public int get_Value();
```

LIBFLAGS Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public enum LIBFLAGS
```

Members

LIBFLAG_FCONTROL

LIBFLAG_FHASDISKIMAGE

LIBFLAG_FHIDDEN

LIBFLAG_FRESTRICTED

Marshal Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Security.SuppressUnmanagedCodeSecurityAttribute() */
public class Marshal
   extends System.Object
```

Public Methods

```
public static int AddRef(
   System.IntPtr pUnk
);
public static System.IntPtr AllocCoTaskMem(
   int cb
);
public static System.IntPtr AllocHGlobal(
   int cb
);
public static System.IntPtr AllocHGlobal(
   System.IntPtr cb
);
public static System.Object BindToMoniker(
   System.String monikerName
);
public static void ChangeWrapperHandleStrength(
   System.Object otp,
   boolean fIsWeak
);
public static void Copy(
   ubyte[] source,
   int startIndex,
   System.IntPtr destination,
   int length
);
public static void Copy(
   char[] source,
   int startIndex,
   System.IntPtr destination,
   int length
);
public static void Copy(
   double[] source,
   int startIndex,
   System.IntPtr destination,
   int length
);
public static void Copy(
   short[] source,
   int startIndex,
   System.IntPtr destination,
   int length
);
public static void Copy(
   int[] source,
   int startIndex,
   System.IntPtr destination,
   int length
);
public static void Copy(
   long[] source,
```

```
      int startIndex,
      System.IntPtr destination,
      int length
   );
   public static void Copy(
      System.IntPtr source,
      ubyte[] destination,
      int startIndex,
      int length
   );
   public static void Copy(
      System.IntPtr source,
      char[] destination,
      int startIndex,
      int length
   );
   public static void Copy(
      System.IntPtr source,
      double[] destination,
      int startIndex,
      int length
   );
   public static void Copy(
      System.IntPtr source,
      short[] destination,
      int startIndex,
      int length
   );
   public static void Copy(
      System.IntPtr source,
      int[] destination,
      int startIndex,
      int length
   );
   public static void Copy(
      System.IntPtr source,
      long[] destination,
      int startIndex,
      int length
   );
   public static void Copy(
      System.IntPtr source,
      float[] destination,
      int startIndex,
      int length
   );
   public static void Copy(
      float[] source,
      int startIndex,
      System.IntPtr destination,
      int length
   );
   public static System.Object CreateWrapperOfType(
      System.Object o,
      System.Type t
   );
   public static void DestroyStructure(
      System.IntPtr ptr,
      System.Type structuretype
   );
   public static void FreeBSTR(
      System.IntPtr ptr
   );
   public static void FreeCoTaskMem(
      System.IntPtr ptr
   );
   public static void FreeHGlobal(
      System.IntPtr hglobal
```

```
);
public static System.Guid GenerateGuidForType(
   System.Type type
);
public static System.String GenerateProgIdForType(
   System.Type type
);
public static System.Object GetActiveObject(
   System.String progID
);
public static System.IntPtr GetComInterfaceForObject(
   System.Object o,
   System.Type T
);
public static System.Object GetComObjectData(
   System.Object obj,
   System.Object key
);
public static int GetComSlotForMethodInfo(
   System.Reflection.MemberInfo m
);
public static int GetEndComSlot(
   System.Type t
);
public static int GetExceptionCode();
public static System.IntPtr GetExceptionPointers();
public static System.IntPtr GetHINSTANCE(
   System.Reflection.Module m
);
public static int GetHRForException(
   System.Exception e
);
public static int GetHRForLastWin32Error();
public static System.IntPtr GetIDispatchForObject(
   System.Object o
);
public static System.IntPtr GetITypeInfoForType(
   System.Type t
);
public static System.IntPtr GetIUnknownForObject(
   System.Object o
);
public static int GetLastWin32Error();
public static System.IntPtr GetManagedThunkForUnmanagedMethodPtr(
   System.IntPtr pfnMethodToWrap,
   System.IntPtr pbSignature,
   int cbSignature
);
public static System.Reflection.MemberInfo GetMethodInfoForComSlot(
   System.Type t,
   int slot,
   System.Runtime.InteropServices.ComMemberType& memberType
);
public static void GetNativeVariantForObject(
   System.Object obj,
   System.IntPtr pDstNativeVariant
);
public static System.Object GetObjectForIUnknown(
   System.IntPtr pUnk
);
public static System.Object GetObjectForNativeVariant(
   System.IntPtr pSrcNativeVariant
);
public static System.Object[] GetObjectsForNativeVariants(
   System.IntPtr aSrcNativeVariant,
   int cVars
);
public static int GetStartComSlot(
```

```
   System.Type t
);
public static System.Threading.Thread GetThreadFromFiberCookie(
   int cookie
);
public static System.Object GetTypedObjectForIUnknown(
   System.IntPtr pUnk,
   System.Type t
);
public static System.Type GetTypeForITypeInfo(
   System.IntPtr piTypeInfo
);
public static System.String GetTypeInfoName(
   System.Runtime.InteropServices.UCOMITypeInfo pTI
);
public static System.Guid GetTypeLibGuid(
   System.Runtime.InteropServices.UCOMITypeLib pTLB
);
public static System.Guid GetTypeLibGuidForAssembly(
   System.Reflection.Assembly asm
);
public static int GetTypeLibLcid(
   System.Runtime.InteropServices.UCOMITypeLib pTLB
);
public static System.String GetTypeLibName(
   System.Runtime.InteropServices.UCOMITypeLib pTLB
);
public static System.IntPtr GetUnmanagedThunkForManagedMethodPtr(
   System.IntPtr pfnMethodToWrap,
   System.IntPtr pbSignature,
   int cbSignature
);
public static boolean IsComObject(
   System.Object o
);
public static boolean IsTypeVisibleFromCom(
   System.Type t
);
public static int NumParamBytes(
   System.Reflection.MethodInfo m
);
public static System.IntPtr OffsetOf(
   System.Type t,
   System.String fieldName
);
public static void Prelink(
   System.Reflection.MethodInfo m
);
public static void PrelinkAll(
   System.Type c
);
public static System.String PtrToStringAnsi(
   System.IntPtr ptr
);
public static System.String PtrToStringAnsi(
   System.IntPtr ptr,
   int len
);
public static System.String PtrToStringAuto(
   System.IntPtr ptr
);
public static System.String PtrToStringAuto(
   System.IntPtr ptr,
   int len
);
public static System.String PtrToStringBSTR(
   System.IntPtr ptr
);
```

```
public static System.String PtrToStringUni(
   System.IntPtr ptr
);
public static System.String PtrToStringUni(
   System.IntPtr ptr,
   int len
);
public static System.Object PtrToStructure(
   System.IntPtr ptr,
   System.Type structureType
);
public static void PtrToStructure(
   System.IntPtr ptr,
   System.Object structure
);
public static int QueryInterface(
   System.IntPtr pUnk,
   System.Guid& iid,
   System.IntPtr& ppv
);
public static ubyte ReadByte(
   System.IntPtr ptr
);
public static ubyte ReadByte(
   System.IntPtr ptr,
   int ofs
);
public static ubyte ReadByte(
   System.Object ptr,
   int ofs
);
public static short ReadInt16(
   System.IntPtr ptr
);
public static short ReadInt16(
   System.IntPtr ptr,
   int ofs
);
public static short ReadInt16(
   System.Object ptr,
   int ofs
);
public static int ReadInt32(
   System.IntPtr ptr
);
public static int ReadInt32(
   System.IntPtr ptr,
   int ofs
);
public static int ReadInt32(
   System.Object ptr,
   int ofs
);
public static long ReadInt64(
   System.IntPtr ptr
);
public static long ReadInt64(
   System.IntPtr ptr,
   int ofs
);
public static long ReadInt64(
   System.Object ptr,
   int ofs
);
public static System.IntPtr ReadIntPtr(
   System.IntPtr ptr
);
public static System.IntPtr ReadIntPtr(
```

```
   System.IntPtr ptr,
   int ofs
);
public static System.IntPtr ReadIntPtr(
   System.Object ptr,
   int ofs
);
public static System.IntPtr ReAllocCoTaskMem(
   System.IntPtr pv,
   int cb
);
public static System.IntPtr ReAllocHGlobal(
   System.IntPtr pv,
   System.IntPtr cb
);
public static int Release(
   System.IntPtr pUnk
);
public static int ReleaseComObject(
   System.Object o
);
public static void ReleaseThreadCache();
public static boolean SetComObjectData(
   System.Object obj,
   System.Object key,
   System.Object data
);
public static int SizeOf(
   System.Object structure
);
public static int SizeOf(
   System.Type t
);
public static System.IntPtr StringToBSTR(
   System.String s
);
public static System.IntPtr StringToCoTaskMemAnsi(
   System.String s
);
public static System.IntPtr StringToCoTaskMemAuto(
   System.String s
);
public static System.IntPtr StringToCoTaskMemUni(
   System.String s
);
public static System.IntPtr StringToHGlobalAnsi(
   System.String s
);
public static System.IntPtr StringToHGlobalAuto(
   System.String s
);
public static System.IntPtr StringToHGlobalUni(
   System.String s
);
public static void StructureToPtr(
   System.Object structure,
   System.IntPtr ptr,
   boolean fDeleteOld
);
public static void ThrowExceptionForHR(
   int errorCode
);
public static void ThrowExceptionForHR(
   int errorCode,
   System.IntPtr errorInfo
);
public static System.IntPtr UnsafeAddrOfPinnedArrayElement(
   System.Array arr,
```

```
    int index
);
public static void WriteByte(
    System.IntPtr ptr,
    ubyte val
);
public static void WriteByte(
    System.IntPtr ptr,
    int ofs,
    ubyte val
);
public static void WriteByte(
    System.Object ptr,
    int ofs,
    ubyte val
);
public static void WriteInt16(
    System.IntPtr ptr,
    char val
);
public static void WriteInt16(
    System.IntPtr ptr,
    short val
);
public static void WriteInt16(
    System.IntPtr ptr,
    int ofs,
    char val
);
public static void WriteInt16(
    System.IntPtr ptr,
    int ofs,
    short val
);
public static void WriteInt16(
    System.Object ptr,
    int ofs,
    char val
);
public static void WriteInt16(
    System.Object ptr,
    int ofs,
    short val
);
public static void WriteInt32(
    System.IntPtr ptr,
    int val
);
public static void WriteInt32(
    System.IntPtr ptr,
    int ofs,
    int val
);
public static void WriteInt32(
    System.Object ptr,
    int ofs,
    int val
);
public static void WriteInt64(
    System.IntPtr ptr,
    long val
);
public static void WriteInt64(
    System.IntPtr ptr,
    int ofs,
    long val
);
public static void WriteInt64(
```

```
      System.Object ptr,
      int ofs,
      long val
   );
   public static void WriteIntPtr(
      System.IntPtr ptr,
      System.IntPtr val
   );
   public static void WriteIntPtr(
      System.IntPtr ptr,
      int ofs,
      System.IntPtr val
   );
   public static void WriteIntPtr(
      System.Object ptr,
      int ofs,
      System.IntPtr val
   );
```

Public Fields

```
   public static final int SystemDefaultCharSize;
   public static final int SystemMaxDBCSCharSize;
```

MarshalAsAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, Parameter, ReturnValue, AllowM
ultiple = false, Inherited = false) */
public class MarshalAsAttribute
   extends System.Attribute
```

Public Constructors

```
public MarshalAsAttribute(
   short unmanagedType
);
public MarshalAsAttribute(
   System.Runtime.InteropServices.UnmanagedType unmanagedType
);
```

Public Properties

```
public System.Runtime.InteropServices.UnmanagedType get_Value();
```

Public Fields

```
public System.Runtime.InteropServices.UnmanagedType ArraySubType;
public System.String MarshalCookie;
public System.String MarshalType;
public System.Type MarshalTypeRef;
public System.Runtime.InteropServices.VarEnum SafeArraySubType;
public System.Type SafeArrayUserDefinedSubType;
public int SizeConst;
public short SizeParamIndex;
```

MarshalDirectiveException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class MarshalDirectiveException
   extends System.SystemException
```

Public Constructors

```
public MarshalDirectiveException();
public MarshalDirectiveException(
   System.String message
);
public MarshalDirectiveException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected MarshalDirectiveException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ObjectCreationDelegate Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
System.IntPtr ObjectCreationDelegate(
   System.IntPtr aggregator
);
```

OptionalAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Parameter, AllowMultiple = false, Inh
erited = false) */
public class OptionalAttribute
   extends System.Attribute
```

Public Constructors

```
public OptionalAttribute();
```

OutAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Parameter, AllowMultiple = false, Inh
erited = false) */
public class OutAttribute
   extends System.Attribute
```

Public Constructors

```
public OutAttribute();
```

PARAMDESC Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class PARAMDESC
   extends System.ValueType
```

Public Fields

```
public System.IntPtr lpVarValue;
public System.Runtime.InteropServices.PARAMFLAG wParamFlags;
```

PARAMFLAG Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public enum PARAMFLAG
```

Members

PARAMFLAG_FHASCUSTDATA

PARAMFLAG_FHASDEFAULT

PARAMFLAG_FIN

PARAMFLAG_FLCID

PARAMFLAG_FOPT

PARAMFLAG_FOUT

PARAMFLAG_FRETVAL

PARAMFLAG_NONE

PreserveSigAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = false) */
public class PreserveSigAttribute
   extends System.Attribute
```

Public Constructors

```
public PreserveSigAttribute();
```

PrimaryInteropAssemblyAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = false) */
public class PrimaryInteropAssemblyAttribute
   extends System.Attribute
```

Public Constructors

```
public PrimaryInteropAssemblyAttribute(
   int major,
   int minor
);
```

Public Properties

```
public int get_MajorVersion();
public int get_MinorVersion();
```

ProgIdAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = false) */
public class ProgIdAttribute
   extends System.Attribute
```

Public Constructors

```
public ProgIdAttribute(
   System.String progId
);
```

Public Properties

```
public System.String get_Value();
```

RegistrationServices Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "475E398F-8AFA-43a7-A3BE-
F4EF8D6787C9") */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = None) */
public class RegistrationServices
   extends System.Object
   implements System.Runtime.InteropServices.IRegistrationServices
```

Public Constructors

```
public RegistrationServices();
```

Public Methods

```
public System.Guid GetManagedCategoryGuid();
public System.String GetProgIdForType(
   System.Type type
);
public System.Type[] GetRegistrableTypesInAssembly(
   System.Reflection.Assembly assembly
);
public boolean RegisterAssembly(
   System.Reflection.Assembly assembly,
   System.Runtime.InteropServices.AssemblyRegistrationFlags flags
);
public void RegisterTypeForComClients(
   System.Type type,
   System.Guid& g
);
public boolean TypeRepresentsComType(
   System.Type type
);
public boolean TypeRequiresRegistration(
   System.Type type
);
public boolean UnregisterAssembly(
   System.Reflection.Assembly assembly
);
```

RuntimeEnvironment Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RuntimeEnvironment
   extends System.Object
```

Public Constructors

```
public RuntimeEnvironment();
```

Public Properties

```
public static System.String get_SystemConfigurationFile();
```

Public Methods

```
public static boolean FromGlobalAccessCache(
   System.Reflection.Assembly a
);
public static System.String GetRuntimeDirectory();
public static System.String GetSystemVersion();
```

SafeArrayRankMismatchException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SafeArrayRankMismatchException
   extends System.SystemException
```

Public Constructors

```
public SafeArrayRankMismatchException();
public SafeArrayRankMismatchException(
   System.String message
);
public SafeArrayRankMismatchException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected SafeArrayRankMismatchException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

SafeArrayTypeMismatchException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SafeArrayTypeMismatchException
   extends System.SystemException
```

Public Constructors

```
public SafeArrayTypeMismatchException();
public SafeArrayTypeMismatchException(
   System.String message
);
public SafeArrayTypeMismatchException(
   System.String message,
   System.Exception inner
);
```

Protected Constructors

```
protected SafeArrayTypeMismatchException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

SEHException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SEHException
   extends System.Runtime.InteropServices.ExternalException
```

Public Constructors

```
public SEHException();
public SEHException(
   System.String message
);
public SEHException(
   System.String message,
   System.Exception inner
);
```

Public Methods

```
public boolean CanResume();
```

Protected Constructors

```
protected SEHException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

STATSTG Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class STATSTG
   extends System.ValueType
```

Public Fields

```
public System.Runtime.InteropServices.FILETIME atime;
public long cbSize;
public System.Guid clsid;
public System.Runtime.InteropServices.FILETIME ctime;
public int grfLocksSupported;
public int grfMode;
public int grfStateBits;
public System.Runtime.InteropServices.FILETIME mtime;
public System.String pwcsName;
public int reserved;
public int type;
```

StructLayoutAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, AllowMultiple = false,
Inherited = false) */
public class StructLayoutAttribute
   extends System.Attribute
```

Public Constructors

```
public StructLayoutAttribute(
   short layoutKind
);
public StructLayoutAttribute(
   System.Runtime.InteropServices.LayoutKind layoutKind
);
```

Public Properties

```
public System.Runtime.InteropServices.LayoutKind get_Value();
```

Public Fields

```
public System.Runtime.InteropServices.CharSet CharSet;
public int Pack;
public int Size;
```

SYSKIND Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public enum SYSKIND
```

Members

SYS_MAC

SYS_WIN16

SYS_WIN32

TYPEATTR Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class TYPEATTR
   extends System.ValueType
```

Public Fields

```
public short cbAlignment;
public int cbSizeInstance;
public short cbSizeVft;
public short cFuncs;
public short cImplTypes;
public short cVars;
public int dwReserved;
public System.Guid guid;
public System.Runtime.InteropServices.IDLDESC idldescType;
public int lcid;
public System.IntPtr lpstrSchema;
public static int MEMBER_ID_NIL;
public int memidConstructor;
public int memidDestructor;
public System.Runtime.InteropServices.TYPEDESC tdescAlias;
public System.Runtime.InteropServices.TYPEKIND typekind;
public short wMajorVerNum;
public short wMinorVerNum;
public System.Runtime.InteropServices.TYPEFLAGS wTypeFlags;
```

TYPEDESC Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class TYPEDESC
   extends System.ValueType
```

Public Fields

```
public System.IntPtr lpValue;
public short vt;
```

TYPEFLAGS Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public enum TYPEFLAGS
```

Members

TYPEFLAG_FAGGREGATABLE

TYPEFLAG_FAPPOBJECT

TYPEFLAG_FCANCREATE

TYPEFLAG_FCONTROL

TYPEFLAG_FDISPATCHABLE

TYPEFLAG_FDUAL

TYPEFLAG_FHIDDEN

TYPEFLAG_FLICENSED

TYPEFLAG_FNONEXTENSIBLE

TYPEFLAG_FOLEAUTOMATION

TYPEFLAG_FPREDECLID

TYPEFLAG_FPROXY

TYPEFLAG_FREPLACEABLE

TYPEFLAG_FRESTRICTED

TYPEFLAG_FREVERSEBIND

TYPEKIND Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public enum TYPEKIND
```

Members

TKIND_ALIAS

TKIND_COCLASS

TKIND_DISPATCH

TKIND_ENUM

TKIND_INTERFACE

TKIND_MAX

TKIND_MODULE

TKIND_RECORD

TKIND_UNION

TYPELIBATTR Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class TYPELIBATTR
   extends System.ValueType
```

Public Fields

```
public System.Guid guid;
public int lcid;
public System.Runtime.InteropServices.SYSKIND syskind;
public System.Runtime.InteropServices.LIBFLAGS wLibFlags;
public short wMajorVerNum;
public short wMinorVerNum;
```

TypeLibConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "F1C3BF79-C3E4-11d3-88E7-
00902754C43A") */
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = None) */
public class TypeLibConverter
   extends System.Object
   implements System.Runtime.InteropServices.ITypeLibConverter
```

Public Constructors

```
public TypeLibConverter();
```

Public Methods

```
public final System.Object ConvertAssemblyToTypeLib(
   System.Reflection.Assembly assembly,
   System.String strTypeLibName,
   System.Runtime.InteropServices.TypeLibExporterFlags flags,
   System.Runtime.InteropServices.ITypeLibExporterNotifySink notifySink
);
public final System.Reflection.Emit.AssemblyBuilder ConvertTypeLibToAssembly(
   System.Object typeLib,
   System.String asmFileName,
   int flags,
   System.Runtime.InteropServices.ITypeLibImporterNotifySink notifySink,
   ubyte[] publicKey,
   System.Reflection.StrongNameKeyPair keyPair,
   boolean unsafeInterfaces
);
public final System.Reflection.Emit.AssemblyBuilder ConvertTypeLibToAssembly(
   System.Object typeLib,
   System.String asmFileName,
   System.Runtime.InteropServices.TypeLibImporterFlags flags,
   System.Runtime.InteropServices.ITypeLibImporterNotifySink notifySink,
   ubyte[] publicKey,
   System.Reflection.StrongNameKeyPair keyPair,
   System.String asmNamespace,
   System.Version asmVersion
);
public final boolean GetPrimaryInteropAssembly(
   System.Guid g,
   int major,
   int minor,
   int lcid,
   System.String& asmName,
   System.String& asmCodeBase
);
```

TypeLibExporterFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum TypeLibExporterFlags
```

Members

OnlyReferenceRegistered

TypeLibFuncAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = false) */
public class TypeLibFuncAttribute
   extends System.Attribute
```

Public Constructors

```
public TypeLibFuncAttribute(
   short flags
);
public TypeLibFuncAttribute(
   System.Runtime.InteropServices.TypeLibFuncFlags flags
);
```

Public Properties

```
public System.Runtime.InteropServices.TypeLibFuncFlags get_Value();
```

TypeLibFuncFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum TypeLibFuncFlags
```

Members

FBindable

FDefaultBind

FDefaultCollelem

FDisplayBind

FHidden

FImmediateBind

FNonBrowsable

FReplaceable

FRequestEdit

FRestricted

FSource

FUiDefault

FUsesGetLastError

TypeLibImporterFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum TypeLibImporterFlags
```

Members

PrimaryInteropAssembly

SafeArrayAsSystemArray

UnsafeInterfaces

TypeLibTypeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, Enum, Interface, Allow
Multiple = false, Inherited = false) */
public class TypeLibTypeAttribute
   extends System.Attribute
```

Public Constructors

```
public TypeLibTypeAttribute(
   short flags
);
public TypeLibTypeAttribute(
   System.Runtime.InteropServices.TypeLibTypeFlags flags
);
```

Public Properties

```
public System.Runtime.InteropServices.TypeLibTypeFlags get_Value();
```

TypeLibTypeFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum TypeLibTypeFlags
```

Members

FAggregatable

FAppObject

FCanCreate

FControl

FDispatchable

FDual

FHidden

FLicensed

FNonExtensible

FOleAutomation

FPreDeclId

FReplaceable

FRestricted

FReverseBind

TypeLibVarAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, AllowMultiple = false, Inherit
ed = false) */
public class TypeLibVarAttribute
   extends System.Attribute
```

Public Constructors

```
public TypeLibVarAttribute(
   short flags
);
public TypeLibVarAttribute(
   System.Runtime.InteropServices.TypeLibVarFlags flags
);
```

Public Properties

```
public System.Runtime.InteropServices.TypeLibVarFlags get_Value();
```

TypeLibVarFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum TypeLibVarFlags
```

Members

FBindable

FDefaultBind

FDefaultCollelem

FDisplayBind

FHidden

FImmediateBind

FNonBrowsable

FReadOnly

FReplaceable

FRequestEdit

FRestricted

FSource

FUiDefault

UCOMIBindCtx Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "0000000e-0000-0000-C000-
000000000046") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface UCOMIBindCtx
```

Public Methods

```
void EnumObjectParam(
   System.Runtime.InteropServices.UCOMIEnumString& ppenum
);
void GetBindOptions(
   System.Runtime.InteropServices.BIND_OPTS& pbindopts
);
void GetObjectParam(
   System.String pszKey,
   System.Object& ppunk
);
void GetRunningObjectTable(
   System.Runtime.InteropServices.UCOMIRunningObjectTable& pprot
);
void RegisterObjectBound(
   System.Object punk
);
void RegisterObjectParam(
   System.String pszKey,
   System.Object punk
);
void ReleaseBoundObjects();
void RevokeObjectBound(
   System.Object punk
);
void RevokeObjectParam(
   System.String pszKey
);
void SetBindOptions(
   System.Runtime.InteropServices.BIND_OPTS& pbindopts
);
```

UCOMIConnectionPoint Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "B196B286-BAB4-101A-B69C-
00AA00341D07") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface UCOMIConnectionPoint
```

Public Methods

```
void Advise(
   System.Object pUnkSink,
   int pdwCookie
);
void EnumConnections(
   System.Runtime.InteropServices.UCOMIEnumConnections& ppEnum
);
void GetConnectionInterface(
   System.Guid& pIID
);
void GetConnectionPointContainer(
   System.Runtime.InteropServices.UCOMIConnectionPointContainer& ppCPC
);
void Unadvise(
   int dwCookie
);
```

UCOMIConnectionPointContainer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "B196B284-BAB4-101A-B69C-
00AA00341D07") */
public interface UCOMIConnectionPointContainer
```

Public Methods

```
void EnumConnectionPoints(
   System.Runtime.InteropServices.UCOMIEnumConnectionPoints& ppEnum
);
void FindConnectionPoint(
   System.Guid& riid,
   System.Runtime.InteropServices.UCOMIConnectionPoint& ppCP
);
```

UCOMIEnumConnectionPoints Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "B196B285-BAB4-101A-B69C-
00AA00341D07") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface UCOMIEnumConnectionPoints
```

Public Methods

```
void Clone(
   System.Runtime.InteropServices.UCOMIEnumConnectionPoints& ppenum
);
int Next(
   int celt,
   System.Runtime.InteropServices.UCOMIConnectionPoint[] rgelt,
   int pceltFetched
);
int Reset();
int Skip(
   int celt
);
```

UCOMIEnumConnections Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "B196B287-BAB4-101A-B69C-
00AA00341D07") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface UCOMIEnumConnections
```

Public Methods

```
void Clone(
   System.Runtime.InteropServices.UCOMIEnumConnections& ppenum
);
int Next(
   int celt,
   System.Runtime.InteropServices.CONNECTDATA[] rgelt,
   int pceltFetched
);
void Reset();
int Skip(
   int celt
);
```

UCOMIEnumMoniker Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "00000102-0000-0000-C000-
000000000046") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface UCOMIEnumMoniker
```

Public Methods

```
void Clone(
   System.Runtime.InteropServices.UCOMIEnumMoniker& ppenum
);
int Next(
   int celt,
   System.Runtime.InteropServices.UCOMIMoniker[] rgelt,
   int pceltFetched
);
int Reset();
int Skip(
   int celt
);
```

UCOMIEnumString Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "00000101-0000-0000-C000-
000000000046") */
public interface UCOMIEnumString
```

Public Methods

```
void Clone(
   System.Runtime.InteropServices.UCOMIEnumString& ppenum
);
int Next(
   int celt,
   System.String[] rgelt,
   int pceltFetched
);
int Reset();
int Skip(
   int celt
);
```

UCOMIEnumVARIANT Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "00020404-0000-0000-C000-
000000000046") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface UCOMIEnumVARIANT
```

Public Methods

```
void Clone(
   int ppenum
);
int Next(
   int celt,
   int rgvar,
   int pceltFetched
);
int Reset();
int Skip(
   int celt
);
```

UCOMIMoniker Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "0000000f-0000-0000-C000-
000000000046") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface UCOMIMoniker
```

Public Methods

```
void BindToObject(
   System.Runtime.InteropServices.UCOMIBindCtx pbc,
   System.Runtime.InteropServices.UCOMIMoniker pmkToLeft,
   System.Guid& riidResult,
   System.Object& ppvResult
);
void BindToStorage(
   System.Runtime.InteropServices.UCOMIBindCtx pbc,
   System.Runtime.InteropServices.UCOMIMoniker pmkToLeft,
   System.Guid& riid,
   System.Object& ppvObj
);
void CommonPrefixWith(
   System.Runtime.InteropServices.UCOMIMoniker pmkOther,
   System.Runtime.InteropServices.UCOMIMoniker& ppmkPrefix
);
void ComposeWith(
   System.Runtime.InteropServices.UCOMIMoniker pmkRight,
   boolean fOnlyIfNotGeneric,
   System.Runtime.InteropServices.UCOMIMoniker& ppmkComposite
);
void Enum(
   boolean fForward,
   System.Runtime.InteropServices.UCOMIEnumMoniker& ppenumMoniker
);
void GetClassID(
   System.Guid& pClassID
);
void GetDisplayName(
   System.Runtime.InteropServices.UCOMIBindCtx pbc,
   System.Runtime.InteropServices.UCOMIMoniker pmkToLeft,
   System.String& ppszDisplayName
);
void GetSizeMax(
   long pcbSize
);
void GetTimeOfLastChange(
   System.Runtime.InteropServices.UCOMIBindCtx pbc,
   System.Runtime.InteropServices.UCOMIMoniker pmkToLeft,
   System.Runtime.InteropServices.FILETIME& pFileTime
);
void Hash(
   int pdwHash
);
void Inverse(
   System.Runtime.InteropServices.UCOMIMoniker& ppmk
);
int IsDirty();
void IsEqual(
```

```
   System.Runtime.InteropServices.UCOMIMoniker pmkOtherMoniker
);
void IsRunning(
   System.Runtime.InteropServices.UCOMIBindCtx pbc,
   System.Runtime.InteropServices.UCOMIMoniker pmkToLeft,
   System.Runtime.InteropServices.UCOMIMoniker pmkNewlyRunning
);
void IsSystemMoniker(
   int pdwMksys
);
void Load(
   System.Runtime.InteropServices.UCOMIStream pStm
);
void ParseDisplayName(
   System.Runtime.InteropServices.UCOMIBindCtx pbc,
   System.Runtime.InteropServices.UCOMIMoniker pmkToLeft,
   System.String pszDisplayName,
   int pchEaten,
   System.Runtime.InteropServices.UCOMIMoniker& ppmkOut
);
void Reduce(
   System.Runtime.InteropServices.UCOMIBindCtx pbc,
   int dwReduceHowFar,
   System.Runtime.InteropServices.UCOMIMoniker& ppmkToLeft,
   System.Runtime.InteropServices.UCOMIMoniker& ppmkReduced
);
void RelativePathTo(
   System.Runtime.InteropServices.UCOMIMoniker pmkOther,
   System.Runtime.InteropServices.UCOMIMoniker& ppmkRelPath
);
void Save(
   System.Runtime.InteropServices.UCOMIStream pStm,
   boolean fClearDirty
);
```

UCOMIPersistFile Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "0000010b-0000-0000-C000-
000000000046") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface UCOMIPersistFile
```

Public Methods

```
void GetClassID(
   System.Guid& pClassID
);
void GetCurFile(
   System.String& ppszFileName
);
int IsDirty();
void Load(
   System.String pszFileName,
   int dwMode
);
void Save(
   System.String pszFileName,
   boolean fRemember
);
void SaveCompleted(
   System.String pszFileName
);
```

UCOMIRunningObjectTable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "00000010-0000-0000-C000-
000000000046") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface UCOMIRunningObjectTable
```

Public Methods

```
void EnumRunning(
   System.Runtime.InteropServices.UCOMIEnumMoniker& ppenumMoniker
);
void GetObject(
   System.Runtime.InteropServices.UCOMIMoniker pmkObjectName,
   System.Object& ppunkObject
);
void GetTimeOfLastChange(
   System.Runtime.InteropServices.UCOMIMoniker pmkObjectName,
   System.Runtime.InteropServices.FILETIME& pfiletime
);
void IsRunning(
   System.Runtime.InteropServices.UCOMIMoniker pmkObjectName
);
void NoteChangeTime(
   int dwRegister,
   System.Runtime.InteropServices.FILETIME& pfiletime
);
void Register(
   int grfFlags,
   System.Object punkObject,
   System.Runtime.InteropServices.UCOMIMoniker pmkObjectName,
   int pdwRegister
);
void Revoke(
   int dwRegister
);
```

UCOMIStream Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "0000000c-0000-0000-C000-
000000000046") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface UCOMIStream
```

Public Methods

```
void Clone(
   System.Runtime.InteropServices.UCOMIStream& ppstm
);
void Commit(
   int grfCommitFlags
);
void CopyTo(
   System.Runtime.InteropServices.UCOMIStream pstm,
   long cb,
   System.IntPtr pcbRead,
   System.IntPtr pcbWritten
);
void LockRegion(
   long libOffset,
   long cb,
   int dwLockType
);
void Read(
   ubyte[] pv,
   int cb,
   System.IntPtr pcbRead
);
void Revert();
void Seek(
   long dlibMove,
   int dwOrigin,
   System.IntPtr plibNewPosition
);
void SetSize(
   long libNewSize
);
void Stat(
   System.Runtime.InteropServices.STATSTG& pstatstg,
   int grfStatFlag
);
void UnlockRegion(
   long libOffset,
   long cb,
   int dwLockType
);
void Write(
   ubyte[] pv,
   int cb,
   System.IntPtr pcbWritten
);
```

UCOMITypeComp Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "00020403-0000-0000-C000-
000000000046") */
public interface UCOMITypeComp
```

Public Methods

```
void Bind(
   System.String szName,
   int lHashVal,
   short wFlags,
   System.Runtime.InteropServices.UCOMITypeInfo& ppTInfo,
   System.Runtime.InteropServices.DESCKIND& pDescKind,
   System.Runtime.InteropServices.BINDPTR& pBindPtr
);
void BindType(
   System.String szName,
   int lHashVal,
   System.Runtime.InteropServices.UCOMITypeInfo& ppTInfo,
   System.Runtime.InteropServices.UCOMITypeComp& ppTComp
);
```

UCOMITypeInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "00020401-0000-0000-C000-
000000000046") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface UCOMITypeInfo
```

Public Methods

```
void AddressOfMember(
   int memid,
   System.Runtime.InteropServices.INVOKEKIND invKind,
   System.IntPtr& ppv
);
void CreateInstance(
   System.Object pUnkOuter,
   System.Guid& riid,
   System.Object& ppvObj
);
void GetContainingTypeLib(
   System.Runtime.InteropServices.UCOMITypeLib& ppTLB,
   int pIndex
);
void GetDllEntry(
   int memid,
   System.Runtime.InteropServices.INVOKEKIND invKind,
   System.String& pBstrDllName,
   System.String& pBstrName,
   short pwOrdinal
);
void GetDocumentation(
   int index,
   System.String& strName,
   System.String& strDocString,
   int dwHelpContext,
   System.String& strHelpFile
);
void GetFuncDesc(
   int index,
   System.IntPtr& ppFuncDesc
);
void GetIDsOfNames(
   System.String[] rgszNames,
   int cNames,
   int[] pMemId
);
void GetImplTypeFlags(
   int index,
   int pImplTypeFlags
);
void GetMops(
   int memid,
   System.String& pBstrMops
);
void GetNames(
   int memid,
   System.String[] rgBstrNames,
   int cMaxNames,
```

```
      int pcNames
   );
   void GetRefTypeInfo(
      int hRef,
      System.Runtime.InteropServices.UCOMITypeInfo& ppTI
   );
   void GetRefTypeOfImplType(
      int index,
      int href
   );
   void GetTypeAttr(
      System.IntPtr& ppTypeAttr
   );
   void GetTypeComp(
      System.Runtime.InteropServices.UCOMITypeComp& ppTComp
   );
   void GetVarDesc(
      int index,
      System.IntPtr& ppVarDesc
   );
   void Invoke(
      System.Object pvInstance,
      int memid,
      short wFlags,
      System.Runtime.InteropServices.DISPPARAMS& pDispParams,
      System.Object& pVarResult,
      System.Runtime.InteropServices.EXCEPINFO& pExcepInfo,
      int puArgErr
   );
   void ReleaseFuncDesc(
      System.IntPtr pFuncDesc
   );
   void ReleaseTypeAttr(
      System.IntPtr pTypeAttr
   );
   void ReleaseVarDesc(
      System.IntPtr pVarDesc
   );
```

UCOMITypeLib Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "00020402-0000-0000-C000-
000000000046") */
public interface UCOMITypeLib
```

Public Methods

```
void FindName(
   System.String szNameBuf,
   int lHashVal,
   System.Runtime.InteropServices.UCOMITypeInfo[] ppTInfo,
   int[] rgMemId,
   short pcFound
);
void GetDocumentation(
   int index,
   System.String& strName,
   System.String& strDocString,
   int dwHelpContext,
   System.String& strHelpFile
);
void GetLibAttr(
   System.IntPtr& ppTLibAttr
);
void GetTypeComp(
   System.Runtime.InteropServices.UCOMITypeComp& ppTComp
);
void GetTypeInfo(
   int index,
   System.Runtime.InteropServices.UCOMITypeInfo& ppTI
);
int GetTypeInfoCount();
void GetTypeInfoOfGuid(
   System.Guid& guid,
   System.Runtime.InteropServices.UCOMITypeInfo& ppTInfo
);
void GetTypeInfoType(
   int index,
   System.Runtime.InteropServices.TYPEKIND& pTKind
);
boolean IsName(
   System.String szNameBuf,
   int lHashVal
);
void ReleaseTLibAttr(
   System.IntPtr pTLibAttr
);
```

UnknownWrapper Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UnknownWrapper
   extends System.Object
```

Public Constructors

```
public UnknownWrapper(
   System.Object obj
);
```

Public Properties

```
public System.Object get_WrappedObject();
```

UnmanagedType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum UnmanagedType
```

Members

AnsiBStr

AsAny

Bool

BStr

ByValArray

ByValTStr

Currency

CustomMarshaler

Error

FunctionPtr

I1

I2

I4

I8

IDispatch

Interface

IUnknown

LPArray

LPStr

LPStruct

LPTStr

LPWStr

R4

R8

SafeArray

Struct

SysInt

SysUInt

TBStr

U1

U2

U4

U8

VariantBool

VBByRefStr

VARDESC Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class VARDESC
   extends System.ValueType
```

Public Fields

```
public System.Runtime.InteropServices.ELEMDESC elemdescVar;
public System.String lpstrSchema;
public int memid;
public System.Runtime.InteropServices.VarEnum varkind;
public short wVarFlags;
```

VARDESC.DESCUNION Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class DESCUNION
   extends System.ValueType
```

Public Fields

```
public System.IntPtr lpvarValue;
public int oInst;
```

VarEnum Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum VarEnum
```

Members

VT_ARRAY

VT_BLOB

VT_BLOB_OBJECT

VT_BOOL

VT_BSTR

VT_BYREF

VT_CARRAY

VT_CF

VT_CLSID

VT_CY

VT_DATE

VT_DECIMAL

VT_DISPATCH

VT_EMPTY

VT_ERROR

VT_FILETIME

VT_HRESULT

VT_I1

VT_I2

VT_I4

VT_I8

VT_INT

VT_LPSTR

VT_LPWSTR

VT_NULL

VT_PTR

VT_R4

VT_R8

VT_RECORD

VT_SAFEARRAY

VT_STORAGE

VT_STORED_OBJECT

VT_STREAM

VT_STREAMED_OBJECT

VT_UI1

VT_UI2

VT_UI4

VT_UI8

VT_UINT

VT_UNKNOWN

VT_USERDEFINED

VT_VARIANT

VT_VECTOR

VT_VOID

VARFLAGS Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public enum VARFLAGS
```

Members

VARFLAG_FBINDABLE

VARFLAG_FDEFAULTBIND

VARFLAG_FDEFAULTCOLLELEM

VARFLAG_FDISPLAYBIND

VARFLAG_FHIDDEN

VARFLAG_FIMMEDIATEBIND

VARFLAG_FNONBROWSABLE

VARFLAG_FREADONLY

VARFLAG_FREPLACEABLE

VARFLAG_FREQUESTEDIT

VARFLAG_FRESTRICTED

VARFLAG_FSOURCE

VARFLAG_FUIDEFAULT

System.Runtime.InteropServices.CustomMarshalers Namespace (Visual J# Syntax)

Classes

EnumerableToDispatchMarshaler

EnumeratorToEnumVariantMarshaler

ExpandoToDispatchExMarshaler

TypeToTypeInfoMarshaler

EnumerableToDispatchMarshaler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EnumerableToDispatchMarshaler
   extends System.Object
   implements System.Runtime.InteropServices.ICustomMarshaler
```

Public Constructors

```
public static EnumerableToDispatchMarshaler();
```

Public Methods

```
public void CleanUpManagedData(
   System.Object pManagedObj
);
public void CleanUpNativeData(
   System.IntPtr pNativeData
);
public static System.Runtime.InteropServices.ICustomMarshaler GetInstance(
   System.String pstrCookie
);
public int GetNativeDataSize();
public System.IntPtr MarshalManagedToNative(
   System.Object pManagedObj
);
public System.Object MarshalNativeToManaged(
   System.IntPtr pNativeData
);
```

EnumeratorToEnumVariantMarshaler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EnumeratorToEnumVariantMarshaler
   extends System.Object
   implements System.Runtime.InteropServices.ICustomMarshaler
```

Public Constructors

```
public static EnumeratorToEnumVariantMarshaler();
```

Public Methods

```
public void CleanUpManagedData(
   System.Object pManagedObj
);
public void CleanUpNativeData(
   System.IntPtr pNativeData
);
public static System.Runtime.InteropServices.ICustomMarshaler GetInstance(
   System.String pstrCookie
);
public int GetNativeDataSize();
public System.IntPtr MarshalManagedToNative(
   System.Object pManagedObj
);
public System.Object MarshalNativeToManaged(
   System.IntPtr pNativeData
);
```

ExpandoToDispatchExMarshaler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ExpandoToDispatchExMarshaler
   extends System.Object
   implements System.Runtime.InteropServices.ICustomMarshaler
```

Public Constructors

```
public static ExpandoToDispatchExMarshaler();
```

Public Methods

```
public void CleanUpManagedData(
   System.Object pManagedObj
);
public void CleanUpNativeData(
   System.IntPtr pNativeData
);
public static System.Runtime.InteropServices.ICustomMarshaler GetInstance(
   System.String pstrCookie
);
public int GetNativeDataSize();
public System.IntPtr MarshalManagedToNative(
   System.Object pManagedObj
);
public System.Object MarshalNativeToManaged(
   System.IntPtr pNativeData
);
```

TypeToTypeInfoMarshaler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TypeToTypeInfoMarshaler
   extends System.Object
   implements System.Runtime.InteropServices.ICustomMarshaler
```

Public Constructors

```
public static TypeToTypeInfoMarshaler();
```

Public Methods

```
public void CleanUpManagedData(
   System.Object pManagedObj
);
public void CleanUpNativeData(
   System.IntPtr pNativeData
);
public static System.Runtime.InteropServices.ICustomMarshaler GetInstance(
   System.String pstrCookie
);
public int GetNativeDataSize();
public System.IntPtr MarshalManagedToNative(
   System.Object pManagedObj
);
public System.Object MarshalNativeToManaged(
   System.IntPtr pNativeData
);
```

System.Runtime.InteropServices.Expando Namespace (Visual J# Syntax)

Interfaces

IExpando

IExpando Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "AFBF15E6-C37C-11d2-B88E-
00A0C9B471B8") */
public interface IExpando
   implements System.Reflection.IReflect
```

Public Methods

```
System.Reflection.FieldInfo AddField(
   System.String name
);
System.Reflection.MethodInfo AddMethod(
   System.String name,
   System.Delegate method
);
System.Reflection.PropertyInfo AddProperty(
   System.String name
);
void RemoveMember(
   System.Reflection.MemberInfo m
);
```

System.Runtime.Remoting Namespace (Visual J# Syntax)

Classes

ActivatedClientTypeEntry

ActivatedServiceTypeEntry

InternalRemotingServices

ObjectHandle

ObjRef

RemotingConfiguration

RemotingException

RemotingServices

RemotingTimeoutException

ServerException

SoapServices

TypeEntry

WellKnownClientTypeEntry

WellKnownServiceTypeEntry

Interfaces

IChannelInfo

IEnvoyInfo

IObjectHandle

IRemotingTypeInfo

Enumerations

WellKnownObjectMode

ActivatedClientTypeEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ActivatedClientTypeEntry
   extends System.Runtime.Remoting.TypeEntry
```

Public Constructors

```
public ActivatedClientTypeEntry(
   System.Type type,
   System.String appUrl
);
public ActivatedClientTypeEntry(
   System.String typeName,
   System.String assemblyName,
   System.String appUrl
);
```

Public Properties

```
public System.String get_ApplicationUrl();
public System.Runtime.Remoting.Contexts.IContextAttribute[] get_ContextAttributes();
public void set_ContextAttributes(
   System.Runtime.Remoting.Contexts.IContextAttribute[] value
);
public System.Type get_ObjectType();
```

Public Methods

```
public System.String ToString();
```

ActivatedServiceTypeEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ActivatedServiceTypeEntry
   extends System.Runtime.Remoting.TypeEntry
```

Public Constructors

```
public ActivatedServiceTypeEntry(
   System.Type type
);
public ActivatedServiceTypeEntry(
   System.String typeName,
   System.String assemblyName
);
```

Public Properties

```
public System.Runtime.Remoting.Contexts.IContextAttribute[] get_ContextAttributes();
public void set_ContextAttributes(
   System.Runtime.Remoting.Contexts.IContextAttribute[] value
);
public System.Type get_ObjectType();
```

Public Methods

```
public System.String ToString();
```

IChannelInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IChannelInfo
```

Public Properties

```
System.Object[] get_ChannelData();
void set_ChannelData(
   System.Object[] value
);
```

IEnvoyInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IEnvoyInfo
```

Public Properties

```
System.Runtime.Remoting.Messaging.IMessageSink get_EnvoySinks();
void set_EnvoySinks(
   System.Runtime.Remoting.Messaging.IMessageSink value
);
```

InternalRemotingServices Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InternalRemotingServices
   extends System.Object
```

Public Constructors

```
public InternalRemotingServices();
```

Public Methods

```
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "_LOGGING") */
public static void DebugOutChnl(
   System.String s
);
public static System.Runtime.Remoting.Metadata.SoapAttribute GetCachedSoapAttribute(
   System.Object reflectionObject
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "_DEBUG") */
public static void RemotingAssert(
   boolean condition,
   System.String message
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "_LOGGING") */
public static void RemotingTrace(
   System.Object[] messages
);
```

IObjectHandle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "C460E2B4-E199-412a-8456-
84DC3E4838C3") */
public interface IObjectHandle
```

Public Methods

```
System.Object Unwrap();
```

IRemotingTypeInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IRemotingTypeInfo
```

Public Properties

```
System.String get_TypeName();
void set_TypeName(
   System.String value
);
```

Public Methods

```
boolean CanCastTo(
   System.Type fromType,
   System.Object o
);
```

ObjectHandle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = AutoDual) */
public class ObjectHandle
   extends System.MarshalByRefObject
   implements System.Runtime.Remoting.IObjectHandle
```

Public Constructors

```
public ObjectHandle(
   System.Object o
);
```

Public Methods

```
public System.Object InitializeLifetimeService();
public final System.Object Unwrap();
```

ObjRef Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ObjRef
   extends System.Object
   implements System.Runtime.Serialization.IObjectReference, System.Runtime.Serialization.ISe
rializable
```

Public Constructors

```
public ObjRef();
public ObjRef(
   System.MarshalByRefObject o,
   System.Type requestedType
);
```

Public Properties

```
public System.Runtime.Remoting.IChannelInfo get_ChannelInfo();
public void set_ChannelInfo(
   System.Runtime.Remoting.IChannelInfo value
);
public System.Runtime.Remoting.IEnvoyInfo get_EnvoyInfo();
public void set_EnvoyInfo(
   System.Runtime.Remoting.IEnvoyInfo value
);
public System.Runtime.Remoting.IRemotingTypeInfo get_TypeInfo();
public void set_TypeInfo(
   System.Runtime.Remoting.IRemotingTypeInfo value
);
public System.String get_URI();
public void set_URI(
   System.String value
);
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public System.Object GetRealObject(
   System.Runtime.Serialization.StreamingContext context
);
public boolean IsFromThisAppDomain();
public boolean IsFromThisProcess();
```

Protected Constructors

```
protected ObjRef(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

RemotingConfiguration Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RemotingConfiguration
   extends System.Object
```

Public Properties

```
public static System.String get_ApplicationId();
public static System.String get_ApplicationName();
public static void set_ApplicationName(
   System.String value
);
public static System.String get_ProcessId();
```

Public Methods

```
public static void Configure(
   System.String filename
);
public static System.Runtime.Remoting.ActivatedClientTypeEntry[] GetRegisteredActivatedClient
Types();
public static System.Runtime.Remoting.ActivatedServiceTypeEntry[] GetRegisteredActivatedServi
ceTypes();
public static System.Runtime.Remoting.WellKnownClientTypeEntry[] GetRegisteredWellKnownClient
Types();
public static System.Runtime.Remoting.WellKnownServiceTypeEntry[] GetRegisteredWellKnownServi
ceTypes();
public static boolean IsActivationAllowed(
   System.Type svrType
);
public static System.Runtime.Remoting.ActivatedClientTypeEntry IsRemotelyActivatedClientType(
   System.Type svrType
);
public static System.Runtime.Remoting.ActivatedClientTypeEntry IsRemotelyActivatedClientType(
   System.String typeName,
   System.String assemblyName
);
public static System.Runtime.Remoting.WellKnownClientTypeEntry IsWellKnownClientType(
   System.Type svrType
);
public static System.Runtime.Remoting.WellKnownClientTypeEntry IsWellKnownClientType(
   System.String typeName,
   System.String assemblyName
);
public static void RegisterActivatedClientType(
   System.Runtime.Remoting.ActivatedClientTypeEntry entry
);
public static void RegisterActivatedClientType(
   System.Type type,
   System.String appUrl
);
public static void RegisterActivatedServiceType(
   System.Runtime.Remoting.ActivatedServiceTypeEntry entry
);
public static void RegisterActivatedServiceType(
   System.Type type
);
```

```
public static void RegisterWellKnownClientType(
   System.Runtime.Remoting.WellKnownClientTypeEntry entry
);
public static void RegisterWellKnownClientType(
   System.Type type,
   System.String objectUrl
);
public static void RegisterWellKnownServiceType(
   System.Runtime.Remoting.WellKnownServiceTypeEntry entry
);
public static void RegisterWellKnownServiceType(
   System.Type type,
   System.String objectUri,
   System.Runtime.Remoting.WellKnownObjectMode mode
);
```

RemotingException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RemotingException
   extends System.SystemException
```

Public Constructors

```
public RemotingException();
public RemotingException(
   System.String message
);
public RemotingException(
   System.String message,
   System.Exception InnerException
);
```

Protected Constructors

```
protected RemotingException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

RemotingServices Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RemotingServices
   extends System.Object
```

Public Methods

```
public static System.Object Connect(
   System.Type classToProxy,
   System.String url
);
public static System.Object Connect(
   System.Type classToProxy,
   System.String url,
   System.Object data
);
public static boolean Disconnect(
   System.MarshalByRefObject obj
);
public static System.Runtime.Remoting.Messaging.IMethodReturnMessage ExecuteMessage(
   System.MarshalByRefObject target,
   System.Runtime.Remoting.Messaging.IMethodCallMessage reqMsg
);
public static System.Runtime.Remoting.Messaging.IMessageSink GetEnvoyChainForProxy(
   System.MarshalByRefObject obj
);
public static System.Object GetLifetimeService(
   System.MarshalByRefObject obj
);
public static System.Reflection.MethodBase GetMethodBaseFromMethodMessage(
   System.Runtime.Remoting.Messaging.IMethodMessage msg
);
public static void GetObjectData(
   System.Object obj,
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public static System.String GetObjectUri(
   System.MarshalByRefObject obj
);
public static System.Runtime.Remoting.ObjRef GetObjRefForProxy(
   System.MarshalByRefObject obj
);
public static System.Runtime.Remoting.Proxies.RealProxy GetRealProxy(
   System.Object proxy
);
public static System.Type GetServerTypeForUri(
   System.String URI
);
public static System.String GetSessionIdForMethodMessage(
   System.Runtime.Remoting.Messaging.IMethodMessage msg
);
public static boolean IsMethodOverloaded(
   System.Runtime.Remoting.Messaging.IMethodMessage msg
);
public static boolean IsObjectOutOfAppDomain(
   System.Object tp
);
public static boolean IsObjectOutOfContext(
```

```
   System.Object tp
);
public static boolean IsOneWay(
   System.Reflection.MethodBase method
);
public static boolean IsTransparentProxy(
   System.Object proxy
);
/** @attribute System.Diagnostics.ConditionalAttribute(ConditionString = "REMOTING_PERF") */
public static void LogRemotingStage(
   int stage
);
public static System.Runtime.Remoting.ObjRef Marshal(
   System.MarshalByRefObject Obj
);
public static System.Runtime.Remoting.ObjRef Marshal(
   System.MarshalByRefObject Obj,
   System.String URI
);
public static System.Runtime.Remoting.ObjRef Marshal(
   System.MarshalByRefObject Obj,
   System.String ObjURI,
   System.Type RequestedType
);
public static void SetObjectUriForMarshal(
   System.MarshalByRefObject obj,
   System.String uri
);
public static System.Object Unmarshal(
   System.Runtime.Remoting.ObjRef objectRef
);
public static System.Object Unmarshal(
   System.Runtime.Remoting.ObjRef objectRef,
   boolean fRefine
);
```

RemotingTimeoutException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RemotingTimeoutException
   extends System.Runtime.Remoting.RemotingException
```

Public Constructors

```
public RemotingTimeoutException();
public RemotingTimeoutException(
   System.String message
);
public RemotingTimeoutException(
   System.String message,
   System.Exception InnerException
);
```

ServerException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ServerException
   extends System.SystemException
```

Public Constructors

```
public ServerException();
public ServerException(
   System.String message
);
public ServerException(
   System.String message,
   System.Exception InnerException
);
```

SoapServices Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapServices
   extends System.Object
```

Public Properties

```
public static System.String get_XmlNsForClrType();
public static System.String get_XmlNsForClrTypeWithAssembly();
public static System.String get_XmlNsForClrTypeWithNs();
public static System.String get_XmlNsForClrTypeWithNsAndAssembly();
```

Public Methods

```
public static System.String CodeXmlNamespaceForClrTypeNamespace(
   System.String typeNamespace,
   System.String assemblyName
);
public static boolean DecodeXmlNamespaceForClrTypeNamespace(
   System.String inNamespace,
   System.String& typeNamespace,
   System.String& assemblyName
);
public static void GetInteropFieldTypeAndNameFromXmlAttribute(
   System.Type containingType,
   System.String xmlAttribute,
   System.String xmlNamespace,
   System.Type& type,
   System.String& name
);
public static void GetInteropFieldTypeAndNameFromXmlElement(
   System.Type containingType,
   System.String xmlElement,
   System.String xmlNamespace,
   System.Type& type,
   System.String& name
);
public static System.Type GetInteropTypeFromXmlElement(
   System.String xmlElement,
   System.String xmlNamespace
);
public static System.Type GetInteropTypeFromXmlType(
   System.String xmlType,
   System.String xmlTypeNamespace
);
public static System.String GetSoapActionFromMethodBase(
   System.Reflection.MethodBase mb
);
public static boolean GetTypeAndMethodNameFromSoapAction(
   System.String soapAction,
   System.String& typeName,
   System.String& methodName
);
public static boolean GetXmlElementForInteropType(
   System.Type type,
   System.String& xmlElement,
   System.String& xmlNamespace
```

```
);
public static System.String GetXmlNamespaceForMethodCall(
   System.Reflection.MethodBase mb
);
public static System.String GetXmlNamespaceForMethodResponse(
   System.Reflection.MethodBase mb
);
public static boolean GetXmlTypeForInteropType(
   System.Type type,
   System.String& xmlType,
   System.String& xmlTypeNamespace
);
public static boolean IsClrTypeNamespace(
   System.String namespaceString
);
public static boolean IsSoapActionValidForMethodBase(
   System.String soapAction,
   System.Reflection.MethodBase mb
);
public static void PreLoad(
   System.Reflection.Assembly assembly
);
public static void PreLoad(
   System.Type type
);
public static void RegisterInteropXmlElement(
   System.String xmlElement,
   System.String xmlNamespace,
   System.Type type
);
public static void RegisterInteropXmlType(
   System.String xmlType,
   System.String xmlTypeNamespace,
   System.Type type
);
public static void RegisterSoapActionForMethodBase(
   System.Reflection.MethodBase mb
);
public static void RegisterSoapActionForMethodBase(
   System.Reflection.MethodBase mb,
   System.String soapAction
);
```

TypeEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TypeEntry
   extends System.Object
```

Public Properties

```
public System.String get_AssemblyName();
public void set_AssemblyName(
   System.String value
);
public System.String get_TypeName();
public void set_TypeName(
   System.String value
);
```

Protected Constructors

```
protected TypeEntry();
```

WellKnownClientTypeEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WellKnownClientTypeEntry
   extends System.Runtime.Remoting.TypeEntry
```

Public Constructors

```
public WellKnownClientTypeEntry(
   System.Type type,
   System.String objectUrl
);
public WellKnownClientTypeEntry(
   System.String typeName,
   System.String assemblyName,
   System.String objectUrl
);
```

Public Properties

```
public System.String get_ApplicationUrl();
public void set_ApplicationUrl(
   System.String value
);
public System.Type get_ObjectType();
public System.String get_ObjectUrl();
```

Public Methods

```
public System.String ToString();
```

WellKnownObjectMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum WellKnownObjectMode
```

Members

SingleCall

Singleton

WellKnownServiceTypeEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WellKnownServiceTypeEntry
   extends System.Runtime.Remoting.TypeEntry
```

Public Constructors

```
public WellKnownServiceTypeEntry(
   System.Type type,
   System.String objectUri,
   System.Runtime.Remoting.WellKnownObjectMode mode
);
public WellKnownServiceTypeEntry(
   System.String typeName,
   System.String assemblyName,
   System.String objectUri,
   System.Runtime.Remoting.WellKnownObjectMode mode
);
```

Public Properties

```
public System.Runtime.Remoting.Contexts.IContextAttribute[] get_ContextAttributes();
public void set_ContextAttributes(
   System.Runtime.Remoting.Contexts.IContextAttribute[] value
);
public System.Runtime.Remoting.WellKnownObjectMode get_Mode();
public System.Type get_ObjectType();
public System.String get_ObjectUri();
```

Public Methods

```
public System.String ToString();
```

System.Runtime.Remoting.Activation Namespace (Visual J# Syntax)

Classes

UrlAttribute

Interfaces

IActivator

IConstructionCallMessage

IConstructionReturnMessage

Enumerations

ActivatorLevel

ActivatorLevel Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ActivatorLevel
```

Members

AppDomain

Construction

Context

Machine

Process

IActivator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IActivator
```

Public Properties

```
System.Runtime.Remoting.Activation.ActivatorLevel get_Level();
System.Runtime.Remoting.Activation.IActivator get_NextActivator();
void set_NextActivator(
   System.Runtime.Remoting.Activation.IActivator value
);
```

Public Methods

```
System.Runtime.Remoting.Activation.IConstructionReturnMessage Activate(
   System.Runtime.Remoting.Activation.IConstructionCallMessage msg
);
```

IConstructionCallMessage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IConstructionCallMessage
   implements System.Runtime.Remoting.Messaging.IMethodCallMessage, System.Runtime.Remoting.M
essaging.IMethodMessage, System.Runtime.Remoting.Messaging.IMessage
```

Public Properties

```
System.Type get_ActivationType();
System.String get_ActivationTypeName();
System.Runtime.Remoting.Activation.IActivator get_Activator();
void set_Activator(
   System.Runtime.Remoting.Activation.IActivator value
);
System.Object[] get_CallSiteActivationAttributes();
System.Collections.IList get_ContextProperties();
```

IConstructionReturnMessage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IConstructionReturnMessage
   implements System.Runtime.Remoting.Messaging.IMethodReturnMessage, System.Runtime.Remoting
.Messaging.IMethodMessage, System.Runtime.Remoting.Messaging.IMessage
```

UrlAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class UrlAttribute
   extends System.Runtime.Remoting.Contexts.ContextAttribute
```

Public Constructors

```
public UrlAttribute(
   System.String callsiteURL
);
```

Public Properties

```
public System.String get_UrlValue();
```

Public Methods

```
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public void GetPropertiesForNewContext(
   System.Runtime.Remoting.Activation.IConstructionCallMessage ctorMsg
);
public boolean IsContextOK(
   System.Runtime.Remoting.Contexts.Context ctx,
   System.Runtime.Remoting.Activation.IConstructionCallMessage msg
);
```

System.Runtime.Remoting.Channels Namespace (Visual J# Syntax)

Classes

BaseChannelObjectWithProperties

BaseChannelSinkWithProperties

BaseChannelWithProperties

BinaryClientFormatterSink

BinaryClientFormatterSinkProvider

BinaryServerFormatterSink

BinaryServerFormatterSinkProvider

ChannelDataStore

ChannelServices

ClientChannelSinkStack

CommonTransportKeys

ServerChannelSinkStack

SinkProviderData

SoapClientFormatterSink

SoapClientFormatterSinkProvider

SoapServerFormatterSink

SoapServerFormatterSinkProvider

TransportHeaders

Interfaces

IChannel

IChannelDataStore

IChannelReceiver

IChannelReceiverHook

IChannelSender

IChannelSinkBase

IClientChannelSink

IClientChannelSinkProvider

IClientChannelSinkStack

IClientFormatterSink

IClientFormatterSinkProvider

IClientResponseChannelSinkStack

IServerChannelSink

IServerChannelSinkProvider

IServerChannelSinkStack

IServerFormatterSinkProvider

IServerResponseChannelSinkStack

ITransportHeaders

Enumerations

BinaryServerFormatterSink.Protocol

ServerProcessing

SoapServerFormatterSink.Protocol

BaseChannelObjectWithProperties Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public abstract class BaseChannelObjectWithProperties
   extends System.Object
   implements System.Collections.IDictionary, System.Collections.ICollection, System.Collecti
ons.IEnumerable
```

Public Constructors

```
public BaseChannelObjectWithProperties();
```

Public Properties

```
public int get_Count();
public boolean get_IsFixedSize();
public boolean get_IsReadOnly();
public boolean get_IsSynchronized();
public System.Object get_Item(
   System.Object key
);
public void set_Item(
   System.Object key,
   System.Object value
);
public System.Collections.ICollection get_Keys();
public System.Collections.IDictionary get_Properties();
public System.Object get_SyncRoot();
public System.Collections.ICollection get_Values();
```

Public Methods

```
public void Add(
   System.Object key,
   System.Object value
);
public void Clear();
public boolean Contains(
   System.Object key
);
public void CopyTo(
   System.Array array,
   int index
);
public System.Collections.IDictionaryEnumerator GetEnumerator();
public void Remove(
   System.Object key
);
```

BaseChannelSinkWithProperties Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class BaseChannelSinkWithProperties
   extends System.Runtime.Remoting.Channels.BaseChannelObjectWithProperties
```

Protected Constructors

```
protected BaseChannelSinkWithProperties();
```

BaseChannelWithProperties Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class BaseChannelWithProperties
   extends System.Runtime.Remoting.Channels.BaseChannelObjectWithProperties
```

Public Properties

```
public System.Collections.IDictionary get_Properties();
```

Protected Constructors

```
protected BaseChannelWithProperties();
```

Protected Fields

```
protected System.Runtime.Remoting.Channels.IChannelSinkBase SinksWithProperties;
```

BinaryClientFormatterSink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BinaryClientFormatterSink
   extends System.Object
   implements System.Runtime.Remoting.Channels.IClientFormatterSink, System.Runtime.Remoting.
Messaging.IMessageSink, System.Runtime.Remoting.Channels.IClientChannelSink, System.Runtime.R
emoting.Channels.IChannelSinkBase
```

Public Constructors

```
public BinaryClientFormatterSink(
   System.Runtime.Remoting.Channels.IClientChannelSink nextSink
);
```

Public Properties

```
public final System.Runtime.Remoting.Channels.IClientChannelSink get_NextChannelSink();
public final System.Runtime.Remoting.Messaging.IMessageSink get_NextSink();
public final System.Collections.IDictionary get_Properties();
```

Public Methods

```
public final System.Runtime.Remoting.Messaging.IMessageCtrl AsyncProcessMessage(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Messaging.IMessageSink replySink
);
public final void AsyncProcessRequest(
   System.Runtime.Remoting.Channels.IClientChannelSinkStack sinkStack,
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers,
   System.IO.Stream stream
);
public final void AsyncProcessResponse(
   System.Runtime.Remoting.Channels.IClientResponseChannelSinkStack sinkStack,
   System.Object state,
   System.Runtime.Remoting.Channels.ITransportHeaders headers,
   System.IO.Stream stream
);
public final System.IO.Stream GetRequestStream(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers
);
public final void ProcessMessage(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders requestHeaders,
   System.IO.Stream requestStream,
   System.Runtime.Remoting.Channels.ITransportHeaders& responseHeaders,
   System.IO.Stream& responseStream
);
public final System.Runtime.Remoting.Messaging.IMessage SyncProcessMessage(
   System.Runtime.Remoting.Messaging.IMessage msg
);
```

BinaryClientFormatterSinkProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BinaryClientFormatterSinkProvider
   extends System.Object
   implements System.Runtime.Remoting.Channels.IClientFormatterSinkProvider, System.Runtime.R
emoting.Channels.IClientChannelSinkProvider
```

Public Constructors

```
public BinaryClientFormatterSinkProvider();
public BinaryClientFormatterSinkProvider(
   System.Collections.IDictionary properties,
   System.Collections.ICollection providerData
);
```

Public Properties

```
public final System.Runtime.Remoting.Channels.IClientChannelSinkProvider get_Next();
public final void set_Next(
   System.Runtime.Remoting.Channels.IClientChannelSinkProvider value
);
```

Public Methods

```
public final System.Runtime.Remoting.Channels.IClientChannelSink CreateSink(
   System.Runtime.Remoting.Channels.IChannelSender channel,
   System.String url,
   System.Object remoteChannelData
);
```

BinaryServerFormatterSink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BinaryServerFormatterSink
   extends System.Object
   implements System.Runtime.Remoting.Channels.IServerChannelSink, System.Runtime.Remoting.Ch
annels.IChannelSinkBase
```

Public Constructors

```
public BinaryServerFormatterSink(
   System.Runtime.Remoting.Channels.BinaryServerFormatterSink+Protocol protocol,
   System.Runtime.Remoting.Channels.IServerChannelSink nextSink,
   System.Runtime.Remoting.Channels.IChannelReceiver receiver
);
```

Public Properties

```
public final System.Runtime.Remoting.Channels.IServerChannelSink get_NextChannelSink();
public final System.Collections.IDictionary get_Properties();
```

Public Methods

```
public final void AsyncProcessResponse(
   System.Runtime.Remoting.Channels.IServerResponseChannelSinkStack sinkStack,
   System.Object state,
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers,
   System.IO.Stream stream
);
public final System.IO.Stream GetResponseStream(
   System.Runtime.Remoting.Channels.IServerResponseChannelSinkStack sinkStack,
   System.Object state,
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers
);
public final System.Runtime.Remoting.Channels.ServerProcessing ProcessMessage(
   System.Runtime.Remoting.Channels.IServerChannelSinkStack sinkStack,
   System.Runtime.Remoting.Messaging.IMessage requestMsg,
   System.Runtime.Remoting.Channels.ITransportHeaders requestHeaders,
   System.IO.Stream requestStream,
   System.Runtime.Remoting.Messaging.IMessage& responseMsg,
   System.Runtime.Remoting.Channels.ITransportHeaders& responseHeaders,
   System.IO.Stream& responseStream
);
```

BinaryServerFormatterSink.Protocol Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum Protocol
```

Members

Http

Other

BinaryServerFormatterSinkProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BinaryServerFormatterSinkProvider
   extends System.Object
   implements System.Runtime.Remoting.Channels.IServerFormatterSinkProvider, System.Runtime.R
emoting.Channels.IServerChannelSinkProvider
```

Public Constructors

```
public BinaryServerFormatterSinkProvider();
public BinaryServerFormatterSinkProvider(
   System.Collections.IDictionary properties,
   System.Collections.ICollection providerData
);
```

Public Properties

```
public final System.Runtime.Remoting.Channels.IServerChannelSinkProvider get_Next();
public final void set_Next(
   System.Runtime.Remoting.Channels.IServerChannelSinkProvider value
);
```

Public Methods

```
public final System.Runtime.Remoting.Channels.IServerChannelSink CreateSink(
   System.Runtime.Remoting.Channels.IChannelReceiver channel
);
public final void GetChannelData(
   System.Runtime.Remoting.Channels.IChannelDataStore channelData
);
```

ChannelDataStore Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ChannelDataStore
   extends System.Object
   implements System.Runtime.Remoting.Channels.IChannelDataStore
```

Public Constructors

```
public ChannelDataStore(
   System.String[] channelURIs
);
```

Public Properties

```
public final System.String[] get_ChannelUris();
public void set_ChannelUris(
   System.String[] value
);
public final System.Object get_Item(
   System.Object key
);
public final void set_Item(
   System.Object key,
   System.Object value
);
```

ChannelServices Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ChannelServices
   extends System.Object
```

Public Properties

```
public static System.Runtime.Remoting.Channels.IChannel[] get_RegisteredChannels();
```

Public Methods

```
public static System.Runtime.Remoting.Messaging.IMessageCtrl AsyncDispatchMessage(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Messaging.IMessageSink replySink
);
public static System.Runtime.Remoting.Channels.IServerChannelSink CreateServerChannelSinkChai
n(
   System.Runtime.Remoting.Channels.IServerChannelSinkProvider provider,
   System.Runtime.Remoting.Channels.IChannelReceiver channel
);
public static System.Runtime.Remoting.Channels.ServerProcessing DispatchMessage(
   System.Runtime.Remoting.Channels.IServerChannelSinkStack sinkStack,
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Messaging.IMessage& replyMsg
);
public static System.Runtime.Remoting.Channels.IChannel GetChannel(
   System.String name
);
public static System.Collections.IDictionary GetChannelSinkProperties(
   System.Object obj
);
public static System.String[] GetUrlsForObject(
   System.MarshalByRefObject obj
);
public static void RegisterChannel(
   System.Runtime.Remoting.Channels.IChannel chnl
);
public static System.Runtime.Remoting.Messaging.IMessage SyncDispatchMessage(
   System.Runtime.Remoting.Messaging.IMessage msg
);
public static void UnregisterChannel(
   System.Runtime.Remoting.Channels.IChannel chnl
);
```

ClientChannelSinkStack Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ClientChannelSinkStack
   extends System.Object
   implements System.Runtime.Remoting.Channels.IClientChannelSinkStack, System.Runtime.Remoti
ng.Channels.IClientResponseChannelSinkStack
```

Public Constructors

```
public ClientChannelSinkStack();
public ClientChannelSinkStack(
   System.Runtime.Remoting.Messaging.IMessageSink replySink
);
```

Public Methods

```
public final void AsyncProcessResponse(
   System.Runtime.Remoting.Channels.ITransportHeaders headers,
   System.IO.Stream stream
);
public final void DispatchException(
   System.Exception e
);
public final void DispatchReplyMessage(
   System.Runtime.Remoting.Messaging.IMessage msg
);
public final System.Object Pop(
   System.Runtime.Remoting.Channels.IClientChannelSink sink
);
public final void Push(
   System.Runtime.Remoting.Channels.IClientChannelSink sink,
   System.Object state
);
```

CommonTransportKeys Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CommonTransportKeys
   extends System.Object
```

Public Constructors

```
public CommonTransportKeys();
```

Public Fields

```
public static System.String ConnectionId;
public static System.String IPAddress;
public static System.String RequestUri;
```

IChannel Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IChannel
```

Public Properties

```
System.String get_ChannelName();
int get_ChannelPriority();
```

Public Methods

```
System.String Parse(
   System.String url,
   System.String& objectURI
);
```

IChannelDataStore Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public interface IChannelDataStore
```

Public Properties

```
System.String[] get_ChannelUris();
System.Object get_Item(
   System.Object key
);
void set_Item(
   System.Object key,
   System.Object value
);
```

IChannelReceiver Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IChannelReceiver
   implements System.Runtime.Remoting.Channels.IChannel
```

Public Properties

```
System.Object get_ChannelData();
```

Public Methods

```
System.String[] GetUrlsForUri(
   System.String objectURI
);
void StartListening(
   System.Object data
);
void StopListening(
   System.Object data
);
```

IChannelReceiverHook Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IChannelReceiverHook
```

Public Properties

```
System.String get_ChannelScheme();
System.Runtime.Remoting.Channels.IServerChannelSink get_ChannelSinkChain();
boolean get_WantsToListen();
```

Public Methods

```
void AddHookChannelUri(
   System.String channelUri
);
```

IChannelSender Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IChannelSender
   implements System.Runtime.Remoting.Channels.IChannel
```

Public Methods

```
System.Runtime.Remoting.Messaging.IMessageSink CreateMessageSink(
   System.String url,
   System.Object remoteChannelData,
   System.String& objectURI
);
```

IChannelSinkBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IChannelSinkBase
```

Public Properties

```
System.Collections.IDictionary get_Properties();
```

IClientChannelSink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IClientChannelSink
   implements System.Runtime.Remoting.Channels.IChannelSinkBase
```

Public Properties

```
System.Runtime.Remoting.Channels.IClientChannelSink get_NextChannelSink();
```

Public Methods

```
void AsyncProcessRequest(
   System.Runtime.Remoting.Channels.IClientChannelSinkStack sinkStack,
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers,
   System.IO.Stream stream
);
void AsyncProcessResponse(
   System.Runtime.Remoting.Channels.IClientResponseChannelSinkStack sinkStack,
   System.Object state,
   System.Runtime.Remoting.Channels.ITransportHeaders headers,
   System.IO.Stream stream
);
System.IO.Stream GetRequestStream(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers
);
void ProcessMessage(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders requestHeaders,
   System.IO.Stream requestStream,
   System.Runtime.Remoting.Channels.ITransportHeaders& responseHeaders,
   System.IO.Stream& responseStream
);
```

IClientChannelSinkProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IClientChannelSinkProvider
```

Public Properties

```
System.Runtime.Remoting.Channels.IClientChannelSinkProvider get_Next();
void set_Next(
   System.Runtime.Remoting.Channels.IClientChannelSinkProvider value
);
```

Public Methods

```
System.Runtime.Remoting.Channels.IClientChannelSink CreateSink(
   System.Runtime.Remoting.Channels.IChannelSender channel,
   System.String url,
   System.Object remoteChannelData
);
```

IClientChannelSinkStack Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IClientChannelSinkStack
   implements System.Runtime.Remoting.Channels.IClientResponseChannelSinkStack
```

Public Methods

```
System.Object Pop(
   System.Runtime.Remoting.Channels.IClientChannelSink sink
);
void Push(
   System.Runtime.Remoting.Channels.IClientChannelSink sink,
   System.Object state
);
```

IClientFormatterSink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IClientFormatterSink
   implements System.Runtime.Remoting.Messaging.IMessageSink, System.Runtime.Remoting.Channel
s.IClientChannelSink, System.Runtime.Remoting.Channels.IChannelSinkBase
```

IClientFormatterSinkProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IClientFormatterSinkProvider
   implements System.Runtime.Remoting.Channels.IClientChannelSinkProvider
```

IClientResponseChannelSinkStack Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IClientResponseChannelSinkStack
```

Public Methods

```
void AsyncProcessResponse(
   System.Runtime.Remoting.Channels.ITransportHeaders headers,
   System.IO.Stream stream
);
void DispatchException(
   System.Exception e
);
void DispatchReplyMessage(
   System.Runtime.Remoting.Messaging.IMessage msg
);
```

IServerChannelSink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IServerChannelSink
   implements System.Runtime.Remoting.Channels.IChannelSinkBase
```

Public Properties

```
System.Runtime.Remoting.Channels.IServerChannelSink get_NextChannelSink();
```

Public Methods

```
void AsyncProcessResponse(
   System.Runtime.Remoting.Channels.IServerResponseChannelSinkStack sinkStack,
   System.Object state,
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers,
   System.IO.Stream stream
);
System.IO.Stream GetResponseStream(
   System.Runtime.Remoting.Channels.IServerResponseChannelSinkStack sinkStack,
   System.Object state,
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers
);
System.Runtime.Remoting.Channels.ServerProcessing ProcessMessage(
   System.Runtime.Remoting.Channels.IServerChannelSinkStack sinkStack,
   System.Runtime.Remoting.Messaging.IMessage requestMsg,
   System.Runtime.Remoting.Channels.ITransportHeaders requestHeaders,
   System.IO.Stream requestStream,
   System.Runtime.Remoting.Messaging.IMessage& responseMsg,
   System.Runtime.Remoting.Channels.ITransportHeaders& responseHeaders,
   System.IO.Stream& responseStream
);
```

IServerChannelSinkProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IServerChannelSinkProvider
```

Public Properties

```
System.Runtime.Remoting.Channels.IServerChannelSinkProvider get_Next();
void set_Next(
   System.Runtime.Remoting.Channels.IServerChannelSinkProvider value
);
```

Public Methods

```
System.Runtime.Remoting.Channels.IServerChannelSink CreateSink(
   System.Runtime.Remoting.Channels.IChannelReceiver channel
);
void GetChannelData(
   System.Runtime.Remoting.Channels.IChannelDataStore channelData
);
```

IServerChannelSinkStack Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IServerChannelSinkStack
   implements System.Runtime.Remoting.Channels.IServerResponseChannelSinkStack
```

Public Methods

```
System.Object Pop(
   System.Runtime.Remoting.Channels.IServerChannelSink sink
);
void Push(
   System.Runtime.Remoting.Channels.IServerChannelSink sink,
   System.Object state
);
void ServerCallback(
   System.IAsyncResult ar
);
void Store(
   System.Runtime.Remoting.Channels.IServerChannelSink sink,
   System.Object state
);
void StoreAndDispatch(
   System.Runtime.Remoting.Channels.IServerChannelSink sink,
   System.Object state
);
```

IServerFormatterSinkProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IServerFormatterSinkProvider
   implements System.Runtime.Remoting.Channels.IServerChannelSinkProvider
```

IServerResponseChannelSinkStack Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IServerResponseChannelSinkStack
```

Public Methods

```
void AsyncProcessResponse(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers,
   System.IO.Stream stream
);
System.IO.Stream GetResponseStream(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers
);
```

ITransportHeaders Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public interface ITransportHeaders
```

Public Properties

```
System.Object get_Item(
   System.Object key
);
void set_Item(
   System.Object key,
   System.Object value
);
```

Public Methods

```
System.Collections.IEnumerator GetEnumerator();
```

ServerChannelSinkStack Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ServerChannelSinkStack
   extends System.Object
   implements System.Runtime.Remoting.Channels.IServerChannelSinkStack, System.Runtime.Remoti
ng.Channels.IServerResponseChannelSinkStack
```

Public Constructors

```
public ServerChannelSinkStack();
```

Public Methods

```
public final void AsyncProcessResponse(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers,
   System.IO.Stream stream
);
public final System.IO.Stream GetResponseStream(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers
);
public final System.Object Pop(
   System.Runtime.Remoting.Channels.IServerChannelSink sink
);
public final void Push(
   System.Runtime.Remoting.Channels.IServerChannelSink sink,
   System.Object state
);
public final void ServerCallback(
   System.IAsyncResult ar
);
public final void Store(
   System.Runtime.Remoting.Channels.IServerChannelSink sink,
   System.Object state
);
public final void StoreAndDispatch(
   System.Runtime.Remoting.Channels.IServerChannelSink sink,
   System.Object state
);
```

ServerProcessing Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ServerProcessing
```

Members

Async

Complete

OneWay

SinkProviderData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SinkProviderData
   extends System.Object
```

Public Constructors

```
public SinkProviderData(
   System.String name
);
```

Public Properties

```
public System.Collections.IList get_Children();
public System.String get_Name();
public System.Collections.IDictionary get_Properties();
```

SoapClientFormatterSink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapClientFormatterSink
   extends System.Object
   implements System.Runtime.Remoting.Channels.IClientFormatterSink, System.Runtime.Remoting.
Messaging.IMessageSink, System.Runtime.Remoting.Channels.IClientChannelSink, System.Runtime.R
emoting.Channels.IChannelSinkBase
```

Public Constructors

```
public SoapClientFormatterSink(
   System.Runtime.Remoting.Channels.IClientChannelSink nextSink
);
```

Public Properties

```
public final System.Runtime.Remoting.Channels.IClientChannelSink get_NextChannelSink();
public final System.Runtime.Remoting.Messaging.IMessageSink get_NextSink();
public final System.Collections.IDictionary get_Properties();
```

Public Methods

```
public final System.Runtime.Remoting.Messaging.IMessageCtrl AsyncProcessMessage(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Messaging.IMessageSink replySink
);
public final void AsyncProcessRequest(
   System.Runtime.Remoting.Channels.IClientChannelSinkStack sinkStack,
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers,
   System.IO.Stream stream
);
public final void AsyncProcessResponse(
   System.Runtime.Remoting.Channels.IClientResponseChannelSinkStack sinkStack,
   System.Object state,
   System.Runtime.Remoting.Channels.ITransportHeaders headers,
   System.IO.Stream stream
);
public final System.IO.Stream GetRequestStream(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers
);
public final void ProcessMessage(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders requestHeaders,
   System.IO.Stream requestStream,
   System.Runtime.Remoting.Channels.ITransportHeaders& responseHeaders,
   System.IO.Stream& responseStream
);
public final System.Runtime.Remoting.Messaging.IMessage SyncProcessMessage(
   System.Runtime.Remoting.Messaging.IMessage msg
);
```

SoapClientFormatterSinkProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapClientFormatterSinkProvider
   extends System.Object
   implements System.Runtime.Remoting.Channels.IClientFormatterSinkProvider, System.Runtime.R
emoting.Channels.IClientChannelSinkProvider
```

Public Constructors

```
public SoapClientFormatterSinkProvider();
public SoapClientFormatterSinkProvider(
   System.Collections.IDictionary properties,
   System.Collections.ICollection providerData
);
```

Public Properties

```
public final System.Runtime.Remoting.Channels.IClientChannelSinkProvider get_Next();
public final void set_Next(
   System.Runtime.Remoting.Channels.IClientChannelSinkProvider value
);
```

Public Methods

```
public final System.Runtime.Remoting.Channels.IClientChannelSink CreateSink(
   System.Runtime.Remoting.Channels.IChannelSender channel,
   System.String url,
   System.Object remoteChannelData
);
```

SoapServerFormatterSink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapServerFormatterSink
   extends System.Object
   implements System.Runtime.Remoting.Channels.IServerChannelSink, System.Runtime.Remoting.Ch
annels.IChannelSinkBase
```

Public Constructors

```
public SoapServerFormatterSink(
   System.Runtime.Remoting.Channels.SoapServerFormatterSink+Protocol protocol,
   System.Runtime.Remoting.Channels.IServerChannelSink nextSink,
   System.Runtime.Remoting.Channels.IChannelReceiver receiver
);
```

Public Properties

```
public final System.Runtime.Remoting.Channels.IServerChannelSink get_NextChannelSink();
public final System.Collections.IDictionary get_Properties();
```

Public Methods

```
public final void AsyncProcessResponse(
   System.Runtime.Remoting.Channels.IServerResponseChannelSinkStack sinkStack,
   System.Object state,
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers,
   System.IO.Stream stream
);
public final System.IO.Stream GetResponseStream(
   System.Runtime.Remoting.Channels.IServerResponseChannelSinkStack sinkStack,
   System.Object state,
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers
);
public final System.Runtime.Remoting.Channels.ServerProcessing ProcessMessage(
   System.Runtime.Remoting.Channels.IServerChannelSinkStack sinkStack,
   System.Runtime.Remoting.Messaging.IMessage requestMsg,
   System.Runtime.Remoting.Channels.ITransportHeaders requestHeaders,
   System.IO.Stream requestStream,
   System.Runtime.Remoting.Messaging.IMessage& responseMsg,
   System.Runtime.Remoting.Channels.ITransportHeaders& responseHeaders,
   System.IO.Stream& responseStream
);
```

SoapServerFormatterSink.Protocol Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum Protocol
```

Members

Http

Other

SoapServerFormatterSinkProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapServerFormatterSinkProvider
   extends System.Object
   implements System.Runtime.Remoting.Channels.IServerFormatterSinkProvider, System.Runtime.R
emoting.Channels.IServerChannelSinkProvider
```

Public Constructors

```
public SoapServerFormatterSinkProvider();
public SoapServerFormatterSinkProvider(
   System.Collections.IDictionary properties,
   System.Collections.ICollection providerData
);
```

Public Properties

```
public final System.Runtime.Remoting.Channels.IServerChannelSinkProvider get_Next();
public final void set_Next(
   System.Runtime.Remoting.Channels.IServerChannelSinkProvider value
);
```

Public Methods

```
public final System.Runtime.Remoting.Channels.IServerChannelSink CreateSink(
   System.Runtime.Remoting.Channels.IChannelReceiver channel
);
public final void GetChannelData(
   System.Runtime.Remoting.Channels.IChannelDataStore channelData
);
```

TransportHeaders Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class TransportHeaders
   extends System.Object
   implements System.Runtime.Remoting.Channels.ITransportHeaders
```

Public Constructors

```
public TransportHeaders();
```

Public Properties

```
public final System.Object get_Item(
   System.Object key
);
public final void set_Item(
   System.Object key,
   System.Object value
);
```

Public Methods

```
public final System.Collections.IEnumerator GetEnumerator();
```

System.Runtime.Remoting.Channels.Http Namespace (Visual J# Syntax)

Classes

HttpChannel

HttpClientChannel

HttpRemotingHandler

HttpRemotingHandlerFactory

HttpServerChannel

HttpChannel Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HttpChannel
   extends System.Runtime.Remoting.Channels.BaseChannelWithProperties
   implements System.Runtime.Remoting.Channels.IChannelReceiver, System.Runtime.Remoting.Chan
nels.IChannel, System.Runtime.Remoting.Channels.IChannelSender, System.Runtime.Remoting.Chann
els.IChannelReceiverHook
```

Public Constructors

```
public HttpChannel();
public HttpChannel(
   int port
);
public HttpChannel(
   System.Collections.IDictionary properties,
   System.Runtime.Remoting.Channels.IClientChannelSinkProvider clientSinkProvider,
   System.Runtime.Remoting.Channels.IServerChannelSinkProvider serverSinkProvider
);
```

Public Properties

```
public final System.Object get_ChannelData();
public final System.String get_ChannelName();
public final int get_ChannelPriority();
public final System.String get_ChannelScheme();
public final System.Runtime.Remoting.Channels.IServerChannelSink get_ChannelSinkChain();
public System.Object get_Item(
   System.Object key
);
public void set_Item(
   System.Object key,
   System.Object value
);
public System.Collections.ICollection get_Keys();
public System.Collections.IDictionary get_Properties();
public final boolean get_WantsToListen();
public void set_WantsToListen(
   boolean value
);
```

Public Methods

```
public final void AddHookChannelUri(
   System.String channelUri
);
public final System.Runtime.Remoting.Messaging.IMessageSink CreateMessageSink(
   System.String url,
   System.Object remoteChannelData,
   System.String& objectURI
);
public final System.String[] GetUrlsForUri(
   System.String objectURI
);
```

```
public final System.String Parse(
   System.String url,
   System.String& objectURI
);
public final void StartListening(
   System.Object data
);
public final void StopListening(
   System.Object data
);
```

HttpClientChannel Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HttpClientChannel
   extends System.Runtime.Remoting.Channels.BaseChannelWithProperties
   implements System.Runtime.Remoting.Channels.IChannelSender, System.Runtime.Remoting.Channe
ls.IChannel
```

Public Constructors

```
public HttpClientChannel();
public HttpClientChannel(
   System.Collections.IDictionary properties,
   System.Runtime.Remoting.Channels.IClientChannelSinkProvider sinkProvider
);
public HttpClientChannel(
   System.String name,
   System.Runtime.Remoting.Channels.IClientChannelSinkProvider sinkProvider
);
```

Public Properties

```
public final System.String get_ChannelName();
public final int get_ChannelPriority();
public System.Object get_Item(
   System.Object key
);
public void set_Item(
   System.Object key,
   System.Object value
);
public System.Collections.ICollection get_Keys();
```

Public Methods

```
public System.Runtime.Remoting.Messaging.IMessageSink CreateMessageSink(
   System.String url,
   System.Object remoteChannelData,
   System.String& objectURI
);
public final System.String Parse(
   System.String url,
   System.String& objectURI
);
```

HttpRemotingHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpRemotingHandler
   extends System.Object
   implements System.Web.IHttpHandler
```

Public Constructors

```
public HttpRemotingHandler();
public HttpRemotingHandler(
   System.Type type,
   System.Object srvID
);
```

Public Properties

```
public final boolean get_IsReusable();
```

Public Methods

```
public final void ProcessRequest(
   System.Web.HttpContext context
);
```

HttpRemotingHandlerFactory Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpRemotingHandlerFactory
   extends System.Object
   implements System.Web.IHttpHandlerFactory
```

Public Constructors

```
public HttpRemotingHandlerFactory();
```

Public Methods

```
public final System.Web.IHttpHandler GetHandler(
   System.Web.HttpContext context,
   System.String verb,
   System.String url,
   System.String filePath
);
public final void ReleaseHandler(
   System.Web.IHttpHandler handler
);
```

HttpServerChannel Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HttpServerChannel
   extends System.Runtime.Remoting.Channels.BaseChannelWithProperties
   implements System.Runtime.Remoting.Channels.IChannelReceiver, System.Runtime.Remoting.Chan
nels.IChannel, System.Runtime.Remoting.Channels.IChannelReceiverHook
```

Public Constructors

```
public HttpServerChannel();
public HttpServerChannel(
   int port
);
public HttpServerChannel(
   System.Collections.IDictionary properties,
   System.Runtime.Remoting.Channels.IServerChannelSinkProvider sinkProvider
);
public HttpServerChannel(
   System.String name,
   int port
);
public HttpServerChannel(
   System.String name,
   int port,
   System.Runtime.Remoting.Channels.IServerChannelSinkProvider sinkProvider
);
```

Public Properties

```
public final System.Object get_ChannelData();
public final System.String get_ChannelName();
public final int get_ChannelPriority();
public final System.String get_ChannelScheme();
public final System.Runtime.Remoting.Channels.IServerChannelSink get_ChannelSinkChain();
public System.Object get_Item(
   System.Object key
);
public void set_Item(
   System.Object key,
   System.Object value
);
public System.Collections.ICollection get_Keys();
public final boolean get_WantsToListen();
public void set_WantsToListen(
   boolean value
);
```

Public Methods

```
public final void AddHookChannelUri(
   System.String channelUri
);
public System.String GetChannelUri();
public System.String[] GetUrlsForUri(
```

```
      System.String objectUri
   );
   public final System.String Parse(
      System.String url,
      System.String& objectURI
   );
   public final void StartListening(
      System.Object data
   );
   public final void StopListening(
      System.Object data
   );
```

System.Runtime.Remoting.Channels.Tcp Namespace (Visual J# Syntax)

Classes

TcpChannel

TcpClientChannel

TcpServerChannel

TcpChannel Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TcpChannel
   extends System.Object
   implements System.Runtime.Remoting.Channels.IChannelReceiver, System.Runtime.Remoting.Chan
nels.IChannel, System.Runtime.Remoting.Channels.IChannelSender
```

Public Constructors

```
public TcpChannel();
public TcpChannel(
   int port
);
public TcpChannel(
   System.Collections.IDictionary properties,
   System.Runtime.Remoting.Channels.IClientChannelSinkProvider clientSinkProvider,
   System.Runtime.Remoting.Channels.IServerChannelSinkProvider serverSinkProvider
);
```

Public Properties

```
public final System.Object get_ChannelData();
public final System.String get_ChannelName();
public final int get_ChannelPriority();
```

Public Methods

```
public final System.Runtime.Remoting.Messaging.IMessageSink CreateMessageSink(
   System.String url,
   System.Object remoteChannelData,
   System.String& objectURI
);
public final System.String[] GetUrlsForUri(
   System.String objectURI
);
public final System.String Parse(
   System.String url,
   System.String& objectURI
);
public final void StartListening(
   System.Object data
);
public final void StopListening(
   System.Object data
);
```

TcpClientChannel Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TcpClientChannel
   extends System.Object
   implements System.Runtime.Remoting.Channels.IChannelSender, System.Runtime.Remoting.Channe
ls.IChannel
```

Public Constructors

```
public TcpClientChannel();
public TcpClientChannel(
   System.Collections.IDictionary properties,
   System.Runtime.Remoting.Channels.IClientChannelSinkProvider sinkProvider
);
public TcpClientChannel(
   System.String name,
   System.Runtime.Remoting.Channels.IClientChannelSinkProvider sinkProvider
);
```

Public Properties

```
public final System.String get_ChannelName();
public final int get_ChannelPriority();
```

Public Methods

```
public System.Runtime.Remoting.Messaging.IMessageSink CreateMessageSink(
   System.String url,
   System.Object remoteChannelData,
   System.String& objectURI
);
public final System.String Parse(
   System.String url,
   System.String& objectURI
);
```

TcpServerChannel Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TcpServerChannel
   extends System.Object
   implements System.Runtime.Remoting.Channels.IChannelReceiver, System.Runtime.Remoting.Chan
nels.IChannel
```

Public Constructors

```
public TcpServerChannel(
   int port
);
public TcpServerChannel(
   System.Collections.IDictionary properties,
   System.Runtime.Remoting.Channels.IServerChannelSinkProvider sinkProvider
);
public TcpServerChannel(
   System.String name,
   int port
);
public TcpServerChannel(
   System.String name,
   int port,
   System.Runtime.Remoting.Channels.IServerChannelSinkProvider sinkProvider
);
```

Public Properties

```
public final System.Object get_ChannelData();
public final System.String get_ChannelName();
public final int get_ChannelPriority();
```

Public Methods

```
public System.String GetChannelUri();
public System.String[] GetUrlsForUri(
   System.String objectUri
);
public final System.String Parse(
   System.String url,
   System.String& objectURI
);
public final void StartListening(
   System.Object data
);
public final void StopListening(
   System.Object data
);
```

System.Runtime.Remoting.Contexts Namespace (Visual J# Syntax)

Classes

Context

ContextAttribute

ContextProperty

SynchronizationAttribute

Interfaces

IContextAttribute

IContextProperty

IContextPropertyActivator

IContributeClientContextSink

IContributeDynamicSink

IContributeEnvoySink

IContributeObjectSink

IContributeServerContextSink

IDynamicMessageSink

IDynamicProperty

Delegates

CrossContextDelegate

Context Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Context
   extends System.Object
```

Public Constructors

```
public Context();
```

Public Properties

```
public int get_ContextID();
public System.Runtime.Remoting.Contexts.IContextProperty[] get_ContextProperties();
public static System.Runtime.Remoting.Contexts.Context get_DefaultContext();
```

Public Methods

```
public static System.LocalDataStoreSlot AllocateDataSlot();
public static System.LocalDataStoreSlot AllocateNamedDataSlot(
   System.String name
);
public void DoCallBack(
   System.Runtime.Remoting.Contexts.CrossContextDelegate deleg
);
public static void FreeNamedDataSlot(
   System.String name
);
public void Freeze();
public static System.Object GetData(
   System.LocalDataStoreSlot slot
);
public static System.LocalDataStoreSlot GetNamedDataSlot(
   System.String name
);
public System.Runtime.Remoting.Contexts.IContextProperty GetProperty(
   System.String name
);
public static boolean RegisterDynamicProperty(
   System.Runtime.Remoting.Contexts.IDynamicProperty prop,
   System.ContextBoundObject obj,
   System.Runtime.Remoting.Contexts.Context ctx
);
public static void SetData(
   System.LocalDataStoreSlot slot,
   System.Object data
);
public void SetProperty(
   System.Runtime.Remoting.Contexts.IContextProperty prop
);
public System.String ToString();
public static boolean UnregisterDynamicProperty(
   System.String name,
   System.ContextBoundObject obj,
   System.Runtime.Remoting.Contexts.Context ctx
);
```

Protected Methods

```
protected void Finalize();
```

ContextAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class ContextAttribute
   extends System.Attribute
   implements System.Runtime.Remoting.Contexts.IContextAttribute, System.Runtime.Remoting.Con
texts.IContextProperty
```

Public Constructors

```
public ContextAttribute(
   System.String name
);
```

Public Properties

```
public System.String get_Name();
```

Public Methods

```
public boolean Equals(
   System.Object o
);
public void Freeze(
   System.Runtime.Remoting.Contexts.Context newContext
);
public int GetHashCode();
public void GetPropertiesForNewContext(
   System.Runtime.Remoting.Activation.IConstructionCallMessage ctorMsg
);
public boolean IsContextOK(
   System.Runtime.Remoting.Contexts.Context ctx,
   System.Runtime.Remoting.Activation.IConstructionCallMessage ctorMsg
);
public boolean IsNewContextOK(
   System.Runtime.Remoting.Contexts.Context newCtx
);
```

Protected Fields

```
protected System.String AttributeName;
```

ContextProperty Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ContextProperty
   extends System.Object
```

Public Properties

```
public System.String get_Name();
public System.Object get_Property();
```

CrossContextDelegate Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void CrossContextDelegate();
```

IContextAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IContextAttribute
```

Public Methods

```
void GetPropertiesForNewContext(
   System.Runtime.Remoting.Activation.IConstructionCallMessage msg
);
boolean IsContextOK(
   System.Runtime.Remoting.Contexts.Context ctx,
   System.Runtime.Remoting.Activation.IConstructionCallMessage msg
);
```

IContextProperty Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IContextProperty
```

Public Properties

```
System.String get_Name();
```

Public Methods

```
void Freeze(
   System.Runtime.Remoting.Contexts.Context newContext
);
boolean IsNewContextOK(
   System.Runtime.Remoting.Contexts.Context newCtx
);
```

IContextPropertyActivator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IContextPropertyActivator
```

Public Methods

```
void CollectFromClientContext(
   System.Runtime.Remoting.Activation.IConstructionCallMessage msg
);
void CollectFromServerContext(
   System.Runtime.Remoting.Activation.IConstructionReturnMessage msg
);
boolean DeliverClientContextToServerContext(
   System.Runtime.Remoting.Activation.IConstructionCallMessage msg
);
boolean DeliverServerContextToClientContext(
   System.Runtime.Remoting.Activation.IConstructionReturnMessage msg
);
boolean IsOKToActivate(
   System.Runtime.Remoting.Activation.IConstructionCallMessage msg
);
```

IContributeClientContextSink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IContributeClientContextSink
```

Public Methods

```
System.Runtime.Remoting.Messaging.IMessageSink GetClientContextSink(
   System.Runtime.Remoting.Messaging.IMessageSink nextSink
);
```

IContributeDynamicSink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IContributeDynamicSink
```

Public Methods

```
System.Runtime.Remoting.Contexts.IDynamicMessageSink GetDynamicSink();
```

IContributeEnvoySink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IContributeEnvoySink
```

Public Methods

```
System.Runtime.Remoting.Messaging.IMessageSink GetEnvoySink(
   System.MarshalByRefObject obj,
   System.Runtime.Remoting.Messaging.IMessageSink nextSink
);
```

IContributeObjectSink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IContributeObjectSink
```

Public Methods

```
System.Runtime.Remoting.Messaging.IMessageSink GetObjectSink(
   System.MarshalByRefObject obj,
   System.Runtime.Remoting.Messaging.IMessageSink nextSink
);
```

IContributeServerContextSink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IContributeServerContextSink
```

Public Methods

```
System.Runtime.Remoting.Messaging.IMessageSink GetServerContextSink(
   System.Runtime.Remoting.Messaging.IMessageSink nextSink
);
```

IDynamicMessageSink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDynamicMessageSink
```

Public Methods

```
void ProcessMessageFinish(
   System.Runtime.Remoting.Messaging.IMessage replyMsg,
   boolean bCliSide,
   boolean bAsync
);
void ProcessMessageStart(
   System.Runtime.Remoting.Messaging.IMessage reqMsg,
   boolean bCliSide,
   boolean bAsync
);
```

IDynamicProperty Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDynamicProperty
```

Public Properties

```
System.String get_Name();
```

SynchronizationAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class SynchronizationAttribute
   extends System.Runtime.Remoting.Contexts.ContextAttribute
   implements System.Runtime.Remoting.Contexts.IContributeServerContextSink, System.Runtime.R
emoting.Contexts.IContributeClientContextSink
```

Public Constructors

```
public SynchronizationAttribute();
public SynchronizationAttribute(
   boolean reEntrant
);
public SynchronizationAttribute(
   int flag
);
public SynchronizationAttribute(
   int flag,
   boolean reEntrant
);
```

Public Properties

```
public boolean get_IsReEntrant();
public boolean get_Locked();
public void set_Locked(
   boolean value
);
```

Public Methods

```
public System.Runtime.Remoting.Messaging.IMessageSink GetClientContextSink(
   System.Runtime.Remoting.Messaging.IMessageSink nextSink
);
public void GetPropertiesForNewContext(
   System.Runtime.Remoting.Activation.IConstructionCallMessage ctorMsg
);
public System.Runtime.Remoting.Messaging.IMessageSink GetServerContextSink(
   System.Runtime.Remoting.Messaging.IMessageSink nextSink
);
public boolean IsContextOK(
   System.Runtime.Remoting.Contexts.Context ctx,
   System.Runtime.Remoting.Activation.IConstructionCallMessage msg
);
```

Public Fields

```
public static int NOT_SUPPORTED;
public static int REQUIRED;
public static int REQUIRES_NEW;
```

```
public static int SUPPORTED;
```

System.Runtime.Remoting.Lifetime Namespace (Visual J# Syntax)

Classes

ClientSponsor

LifetimeServices

Interfaces

ILease

ISponsor

Enumerations

LeaseState

ClientSponsor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ClientSponsor
   extends System.MarshalByRefObject
   implements System.Runtime.Remoting.Lifetime.ISponsor
```

Public Constructors

```
public ClientSponsor();
public ClientSponsor(
   System.TimeSpan renewalTime
);
```

Public Properties

```
public System.TimeSpan get_RenewalTime();
public void set_RenewalTime(
   System.TimeSpan value
);
```

Public Methods

```
public void Close();
public System.Object InitializeLifetimeService();
public boolean Register(
   System.MarshalByRefObject obj
);
public final System.TimeSpan Renewal(
   System.Runtime.Remoting.Lifetime.ILease lease
);
public void Unregister(
   System.MarshalByRefObject obj
);
```

Protected Methods

```
protected void Finalize();
```

ILease Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ILease
```

Public Properties

```
System.TimeSpan get_CurrentLeaseTime();
System.Runtime.Remoting.Lifetime.LeaseState get_CurrentState();
System.TimeSpan get_InitialLeaseTime();
void set_InitialLeaseTime(
   System.TimeSpan value
);
System.TimeSpan get_RenewOnCallTime();
void set_RenewOnCallTime(
   System.TimeSpan value
);
System.TimeSpan get_SponsorshipTimeout();
void set_SponsorshipTimeout(
   System.TimeSpan value
);
```

Public Methods

```
void Register(
   System.Runtime.Remoting.Lifetime.ISponsor obj
);
void Register(
   System.Runtime.Remoting.Lifetime.ISponsor obj,
   System.TimeSpan renewalTime
);
System.TimeSpan Renew(
   System.TimeSpan renewalTime
);
void Unregister(
   System.Runtime.Remoting.Lifetime.ISponsor obj
);
```

ISponsor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISponsor
```

Public Methods

```
System.TimeSpan Renewal(
   System.Runtime.Remoting.Lifetime.ILease lease
);
```

LeaseState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum LeaseState
```

Members

Active

Expired

Initial

Null

Renewing

LifetimeServices Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LifetimeServices
   extends System.Object
```

Public Constructors

```
public LifetimeServices();
```

Public Properties

```
public static System.TimeSpan get_LeaseManagerPollTime();
public static void set_LeaseManagerPollTime(
   System.TimeSpan value
);
public static System.TimeSpan get_LeaseTime();
public static void set_LeaseTime(
   System.TimeSpan value
);
public static System.TimeSpan get_RenewOnCallTime();
public static void set_RenewOnCallTime(
   System.TimeSpan value
);
public static System.TimeSpan get_SponsorshipTimeout();
public static void set_SponsorshipTimeout(
   System.TimeSpan value
);
```

System.Runtime.Remoting.Messaging Namespace (Visual J# Syntax)

Classes

AsyncResult

CallContext

ConstructionCall

ConstructionResponse

Header

InternalMessageWrapper

LogicalCallContext

MethodCall

MethodCallMessageWrapper

MethodResponse

MethodReturnMessageWrapper

OneWayAttribute

RemotingSurrogateSelector

ReturnMessage

Interfaces

ILogicalThreadAffinative

IMessage

IMessageCtrl

IMessageSink

IMethodCallMessage

IMethodMessage

IMethodReturnMessage

IRemotingFormatter

Delegates

HeaderHandler

MessageSurrogateFilter

AsyncResult Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AsyncResult
   extends System.Object
   implements System.IAsyncResult, System.Runtime.Remoting.Messaging.IMessageSink
```

Public Properties

```
public System.Object get_AsyncDelegate();
public System.Object get_AsyncState();
public System.Threading.WaitHandle get_AsyncWaitHandle();
public boolean get_CompletedSynchronously();
public boolean get_EndInvokeCalled();
public void set_EndInvokeCalled(
   boolean value
);
public boolean get_IsCompleted();
public final System.Runtime.Remoting.Messaging.IMessageSink get_NextSink();
```

Public Methods

```
public System.Runtime.Remoting.Messaging.IMessageCtrl AsyncProcessMessage(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Messaging.IMessageSink replySink
);
public System.Runtime.Remoting.Messaging.IMessage GetReplyMessage();
public void SetMessageCtrl(
   System.Runtime.Remoting.Messaging.IMessageCtrl mc
);
public System.Runtime.Remoting.Messaging.IMessage SyncProcessMessage(
   System.Runtime.Remoting.Messaging.IMessage msg
);
```

CallContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class CallContext
   extends System.Object
```

Public Methods

```
public static void FreeNamedDataSlot(
   System.String name
);
public static System.Object GetData(
   System.String name
);
public static System.Runtime.Remoting.Messaging.Header[] GetHeaders();
public static void SetData(
   System.String name,
   System.Object data
);
public static void SetHeaders(
   System.Runtime.Remoting.Messaging.Header[] headers
);
```

ConstructionCall Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public class ConstructionCall
   extends System.Runtime.Remoting.Messaging.MethodCall
   implements System.Runtime.Remoting.Activation.IConstructionCallMessage
```

Public Constructors

```
public ConstructionCall(
   System.Runtime.Remoting.Messaging.Header[] headers
);
public ConstructionCall(
   System.Runtime.Remoting.Messaging.IMessage m
);
```

Public Properties

```
public final System.Type get_ActivationType();
public final System.String get_ActivationTypeName();
public final System.Runtime.Remoting.Activation.IActivator get_Activator();
public final void set_Activator(
   System.Runtime.Remoting.Activation.IActivator value
);
public final System.Object[] get_CallSiteActivationAttributes();
public final System.Collections.IList get_ContextProperties();
public System.Collections.IDictionary get_Properties();
```

ConstructionResponse Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public class ConstructionResponse
   extends System.Runtime.Remoting.Messaging.MethodResponse
   implements System.Runtime.Remoting.Activation.IConstructionReturnMessage
```

Public Constructors

```
public ConstructionResponse(
   System.Runtime.Remoting.Messaging.Header[] h,
   System.Runtime.Remoting.Messaging.IMethodCallMessage mcm
);
```

Public Properties

```
public System.Collections.IDictionary get_Properties();
```

Header Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Header
   extends System.Object
```

Public Constructors

```
public Header(
   System.String _Name,
   System.Object _Value
);
public Header(
   System.String _Name,
   System.Object _Value,
   boolean _MustUnderstand
);
public Header(
   System.String _Name,
   System.Object _Value,
   boolean _MustUnderstand,
   System.String _HeaderNamespace
);
```

Public Fields

```
public System.String HeaderNamespace;
public boolean MustUnderstand;
public System.String Name;
public System.Object Value;
```

HeaderHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
System.Object HeaderHandler(
   System.Runtime.Remoting.Messaging.Header[] headers
);
```

ILogicalThreadAffinative Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ILogicalThreadAffinative
```

IMessage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IMessage
```

Public Properties

```
System.Collections.IDictionary get_Properties();
```

IMessageCtrl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IMessageCtrl
```

Public Methods

```
void Cancel(
   int msToCancel
);
```

IMessageSink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IMessageSink
```

Public Properties

```
System.Runtime.Remoting.Messaging.IMessageSink get_NextSink();
```

Public Methods

```
System.Runtime.Remoting.Messaging.IMessageCtrl AsyncProcessMessage(
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Messaging.IMessageSink replySink
);
System.Runtime.Remoting.Messaging.IMessage SyncProcessMessage(
   System.Runtime.Remoting.Messaging.IMessage msg
);
```

IMethodCallMessage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IMethodCallMessage
   implements System.Runtime.Remoting.Messaging.IMethodMessage, System.Runtime.Remoting.Messa
ging.IMessage
```

Public Properties

```
int get_InArgCount();
System.Object[] get_InArgs();
```

Public Methods

```
System.Object GetInArg(
   int argNum
);
System.String GetInArgName(
   int index
);
```

IMethodMessage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IMethodMessage
   implements System.Runtime.Remoting.Messaging.IMessage
```

Public Properties

```
int get_ArgCount();
System.Object[] get_Args();
boolean get_HasVarArgs();
System.Runtime.Remoting.Messaging.LogicalCallContext get_LogicalCallContext();
System.Reflection.MethodBase get_MethodBase();
System.String get_MethodName();
System.Object get_MethodSignature();
System.String get_TypeName();
System.String get_Uri();
```

Public Methods

```
System.Object GetArg(
   int argNum
);
System.String GetArgName(
   int index
);
```

IMethodReturnMessage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IMethodReturnMessage
   implements System.Runtime.Remoting.Messaging.IMethodMessage, System.Runtime.Remoting.Messa
ging.IMessage
```

Public Properties

```
System.Exception get_Exception();
int get_OutArgCount();
System.Object[] get_OutArgs();
System.Object get_ReturnValue();
```

Public Methods

```
System.Object GetOutArg(
   int argNum
);
System.String GetOutArgName(
   int index
);
```

InternalMessageWrapper Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InternalMessageWrapper
   extends System.Object
```

Public Constructors

```
public InternalMessageWrapper(
   System.Runtime.Remoting.Messaging.IMessage msg
);
```

Protected Fields

```
protected System.Runtime.Remoting.Messaging.IMessage WrappedMessage;
```

IRemotingFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IRemotingFormatter
   implements System.Runtime.Serialization.IFormatter
```

Public Methods

```
System.Object Deserialize(
   System.IO.Stream serializationStream,
   System.Runtime.Remoting.Messaging.HeaderHandler handler
);
void Serialize(
   System.IO.Stream serializationStream,
   System.Object graph,
   System.Runtime.Remoting.Messaging.Header[] headers
);
```

LogicalCallContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class LogicalCallContext
   extends System.Object
   implements System.Runtime.Serialization.ISerializable, System.ICloneable
```

Public Properties

```
public boolean get_HasInfo();
```

Public Methods

```
public final System.Object Clone();
public void FreeNamedDataSlot(
   System.String name
);
public System.Object GetData(
   System.String name
);
public final void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public void SetData(
   System.String name,
   System.Object data
);
```

MessageSurrogateFilter Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
boolean MessageSurrogateFilter(
   System.String key,
   System.Object value
);
```

MethodCall Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public class MethodCall
   extends System.Object
   implements System.Runtime.Remoting.Messaging.IMethodCallMessage, System.Runtime.Remoting.M
essaging.IMethodMessage, System.Runtime.Remoting.Messaging.IMessage, System.Runtime.Serializa
tion.ISerializable, System.Runtime.Remoting.Messaging.IInternalMessage, System.Runtime.Remoti
ng.Messaging.ISerializationRootObject
```

Public Constructors

```
public MethodCall(
   System.Runtime.Remoting.Messaging.Header[] h1
);
public MethodCall(
   System.Runtime.Remoting.Messaging.IMessage msg
);
```

Public Properties

```
public final int get_ArgCount();
public final System.Object[] get_Args();
public final boolean get_HasVarArgs();
public final int get_InArgCount();
public final System.Object[] get_InArgs();
public final System.Runtime.Remoting.Messaging.LogicalCallContext get_LogicalCallContext();
public final System.Reflection.MethodBase get_MethodBase();
public final System.String get_MethodName();
public final System.Object get_MethodSignature();
public System.Collections.IDictionary get_Properties();
public final System.String get_TypeName();
public final System.String get_Uri();
public void set_Uri(
   System.String value
);
```

Public Methods

```
public final System.Object GetArg(
   int argNum
);
public final System.String GetArgName(
   int index
);
public final System.Object GetInArg(
   int argNum
);
public final System.String GetInArgName(
   int index
);
public final void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
```

```
);
public System.Object HeaderHandler(
   System.Runtime.Remoting.Messaging.Header[] h
);
public void Init();
public void ResolveMethod();
public final void RootSetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext ctx
);
```

Protected Fields

```
protected System.Collections.IDictionary ExternalProperties;
protected System.Collections.IDictionary InternalProperties;
```

MethodCallMessageWrapper Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MethodCallMessageWrapper
   extends System.Runtime.Remoting.Messaging.InternalMessageWrapper
   implements System.Runtime.Remoting.Messaging.IMethodCallMessage, System.Runtime.Remoting.M
essaging.IMethodMessage, System.Runtime.Remoting.Messaging.IMessage
```

Public Constructors

```
public MethodCallMessageWrapper(
   System.Runtime.Remoting.Messaging.IMethodCallMessage msg
);
```

Public Properties

```
public int get_ArgCount();
public System.Object[] get_Args();
public void set_Args(
   System.Object[] value
);
public boolean get_HasVarArgs();
public int get_InArgCount();
public System.Object[] get_InArgs();
public System.Runtime.Remoting.Messaging.LogicalCallContext get_LogicalCallContext();
public System.Reflection.MethodBase get_MethodBase();
public System.String get_MethodName();
public System.Object get_MethodSignature();
public System.Collections.IDictionary get_Properties();
public System.String get_TypeName();
public System.String get_Uri();
public void set_Uri(
   System.String value
);
```

Public Methods

```
public System.Object GetArg(
   int argNum
);
public System.String GetArgName(
   int index
);
public System.Object GetInArg(
   int argNum
);
public System.String GetInArgName(
   int index
);
```

MethodResponse Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public class MethodResponse
   extends System.Object
   implements System.Runtime.Remoting.Messaging.IMethodReturnMessage, System.Runtime.Remoting
.Messaging.IMethodMessage, System.Runtime.Remoting.Messaging.IMessage, System.Runtime.Seriali
zation.ISerializable, System.Runtime.Remoting.Messaging.ISerializationRootObject, System.Runt
ime.Remoting.Messaging.IInternalMessage
```

Public Constructors

```
public MethodResponse(
   System.Runtime.Remoting.Messaging.Header[] h1,
   System.Runtime.Remoting.Messaging.IMethodCallMessage mcm
);
```

Public Properties

```
public final int get_ArgCount();
public final System.Object[] get_Args();
public final System.Exception get_Exception();
public final boolean get_HasVarArgs();
public final System.Runtime.Remoting.Messaging.LogicalCallContext get_LogicalCallContext();
public final System.Reflection.MethodBase get_MethodBase();
public final System.String get_MethodName();
public final System.Object get_MethodSignature();
public final int get_OutArgCount();
public final System.Object[] get_OutArgs();
public System.Collections.IDictionary get_Properties();
public final System.Object get_ReturnValue();
public final System.String get_TypeName();
public final System.String get_Uri();
public void set_Uri(
   System.String value
);
```

Public Methods

```
public final System.Object GetArg(
   int argNum
);
public final System.String GetArgName(
   int index
);
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public final System.Object GetOutArg(
   int argNum
);
public final System.String GetOutArgName(
   int index
```

```
);
public System.Object HeaderHandler(
   System.Runtime.Remoting.Messaging.Header[] h
);
public final void RootSetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext ctx
);
```

Protected Fields

```
protected System.Collections.IDictionary ExternalProperties;
protected System.Collections.IDictionary InternalProperties;
```

MethodReturnMessageWrapper Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MethodReturnMessageWrapper
   extends System.Runtime.Remoting.Messaging.InternalMessageWrapper
   implements System.Runtime.Remoting.Messaging.IMethodReturnMessage, System.Runtime.Remoting
.Messaging.IMethodMessage, System.Runtime.Remoting.Messaging.IMessage
```

Public Constructors

```
public MethodReturnMessageWrapper(
   System.Runtime.Remoting.Messaging.IMethodReturnMessage msg
);
```

Public Properties

```
public int get_ArgCount();
public System.Object[] get_Args();
public void set_Args(
   System.Object[] value
);
public System.Exception get_Exception();
public void set_Exception(
   System.Exception value
);
public boolean get_HasVarArgs();
public System.Runtime.Remoting.Messaging.LogicalCallContext get_LogicalCallContext();
public System.Reflection.MethodBase get_MethodBase();
public System.String get_MethodName();
public System.Object get_MethodSignature();
public int get_OutArgCount();
public System.Object[] get_OutArgs();
public System.Collections.IDictionary get_Properties();
public System.Object get_ReturnValue();
public void set_ReturnValue(
   System.Object value
);
public System.String get_TypeName();
public final System.String get_Uri();
public void set_Uri(
   System.String value
);
```

Public Methods

```
public System.Object GetArg(
   int argNum
);
public System.String GetArgName(
   int index
);
public System.Object GetOutArg(
   int argNum
);
public System.String GetOutArgName(
   int index
```

```
);
```

OneWayAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = true) */
public class OneWayAttribute
   extends System.Attribute
```

Public Constructors

```
public OneWayAttribute();
```

RemotingSurrogateSelector Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RemotingSurrogateSelector
   extends System.Object
   implements System.Runtime.Serialization.ISurrogateSelector
```

Public Constructors

```
public RemotingSurrogateSelector();
```

Public Properties

```
public System.Runtime.Remoting.Messaging.MessageSurrogateFilter get_Filter();
public void set_Filter(
   System.Runtime.Remoting.Messaging.MessageSurrogateFilter value
);
```

Public Methods

```
public void ChainSelector(
   System.Runtime.Serialization.ISurrogateSelector selector
);
public System.Runtime.Serialization.ISurrogateSelector GetNextSelector();
public System.Object GetRootObject();
public System.Runtime.Serialization.ISerializationSurrogate GetSurrogate(
   System.Type type,
   System.Runtime.Serialization.StreamingContext context,
   System.Runtime.Serialization.ISurrogateSelector& ssout
);
public void SetRootObject(
   System.Object obj
);
public void UseSoapFormat();
```

ReturnMessage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ReturnMessage
   extends System.Object
   implements System.Runtime.Remoting.Messaging.IMethodReturnMessage, System.Runtime.Remoting
.Messaging.IMethodMessage, System.Runtime.Remoting.Messaging.IMessage
```

Public Constructors

```
public ReturnMessage(
   System.Exception e,
   System.Runtime.Remoting.Messaging.IMethodCallMessage mcm
);
public ReturnMessage(
   System.Object ret,
   System.Object[] outArgs,
   int outArgsCount,
   System.Runtime.Remoting.Messaging.LogicalCallContext callCtx,
   System.Runtime.Remoting.Messaging.IMethodCallMessage mcm
);
```

Public Properties

```
public final int get_ArgCount();
public final System.Object[] get_Args();
public final System.Exception get_Exception();
public final boolean get_HasVarArgs();
public final System.Runtime.Remoting.Messaging.LogicalCallContext get_LogicalCallContext();
public final System.Reflection.MethodBase get_MethodBase();
public final System.String get_MethodName();
public final System.Object get_MethodSignature();
public final int get_OutArgCount();
public final System.Object[] get_OutArgs();
public System.Collections.IDictionary get_Properties();
public System.Object get_ReturnValue();
public final System.String get_TypeName();
public final System.String get_Uri();
public void set_Uri(
   System.String value
);
```

Public Methods

```
public final System.Object GetArg(
   int argNum
);
public final System.String GetArgName(
   int index
);
public final System.Object GetOutArg(
   int argNum
);
public final System.String GetOutArgName(
   int index
);
```

System.Runtime.Remoting.Metadata Namespace (Visual J# Syntax)

Classes

SoapAttribute

SoapFieldAttribute

SoapMethodAttribute

SoapParameterAttribute

SoapTypeAttribute

Enumerations

SoapOption

XmlFieldOrderOption

SoapAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapAttribute
   extends System.Attribute
```

Public Constructors

```
public SoapAttribute();
```

Public Properties

```
public boolean get_Embedded();
public void set_Embedded(
   boolean value
);
public boolean get_UseAttribute();
public void set_UseAttribute(
   boolean value
);
public System.String get_XmlNamespace();
public void set_XmlNamespace(
   System.String value
);
```

Protected Fields

```
protected System.String ProtXmlNamespace;
protected System.Object ReflectInfo;
```

SoapFieldAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, AllowMultiple = false, Inherit
ed = true) */
public class SoapFieldAttribute
   extends System.Runtime.Remoting.Metadata.SoapAttribute
```

Public Constructors

```
public SoapFieldAttribute();
```

Public Properties

```
public int get_Order();
public void set_Order(
   int value
);
public System.String get_XmlElementName();
public void set_XmlElementName(
   System.String value
);
```

Public Methods

```
public boolean IsInteropXmlElement();
```

SoapMethodAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = true) */
public class SoapMethodAttribute
   extends System.Runtime.Remoting.Metadata.SoapAttribute
```

Public Constructors

```
public SoapMethodAttribute();
```

Public Properties

```
public System.String get_ResponseXmlElementName();
public void set_ResponseXmlElementName(
   System.String value
);
public System.String get_ResponseXmlNamespace();
public void set_ResponseXmlNamespace(
   System.String value
);
public System.String get_ReturnXmlElementName();
public void set_ReturnXmlElementName(
   System.String value
);
public System.String get_SoapAction();
public void set_SoapAction(
   System.String value
);
public boolean get_UseAttribute();
public void set_UseAttribute(
   boolean value
);
public System.String get_XmlNamespace();
public void set_XmlNamespace(
   System.String value
);
```

SoapOption Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum SoapOption
```

Members

AlwaysIncludeTypes

EmbedAll

None

Option1

Option2

XsdString

SoapParameterAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Parameter, AllowMultiple = false, Inh
erited = true) */
public class SoapParameterAttribute
   extends System.Runtime.Remoting.Metadata.SoapAttribute
```

Public Constructors

```
public SoapParameterAttribute();
```

SoapTypeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, Enum, Interface, Allow
Multiple = false, Inherited = true) */
public class SoapTypeAttribute
   extends System.Runtime.Remoting.Metadata.SoapAttribute
```

Public Constructors

```
public SoapTypeAttribute();
```

Public Properties

```
public System.Runtime.Remoting.Metadata.SoapOption get_SoapOptions();
public void set_SoapOptions(
   System.Runtime.Remoting.Metadata.SoapOption value
);
public boolean get_UseAttribute();
public void set_UseAttribute(
   boolean value
);
public System.String get_XmlElementName();
public void set_XmlElementName(
   System.String value
);
public System.Runtime.Remoting.Metadata.XmlFieldOrderOption get_XmlFieldOrder();
public void set_XmlFieldOrder(
   System.Runtime.Remoting.Metadata.XmlFieldOrderOption value
);
public System.String get_XmlNamespace();
public void set_XmlNamespace(
   System.String value
);
public System.String get_XmlTypeName();
public void set_XmlTypeName(
   System.String value
);
public System.String get_XmlTypeNamespace();
public void set_XmlTypeNamespace(
   System.String value
);
```

XmlFieldOrderOption Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlFieldOrderOption
```

Members

All

Choice

Sequence

System.Runtime.Remoting.Metadata.W3cXsd2001 Namespace (Visual J# Syntax)

Classes

SoapAnyUri

SoapBase64Binary

SoapDate

SoapDateTime

SoapDay

SoapDuration

SoapEntities

SoapEntity

SoapHexBinary

SoapId

SoapIdref

SoapIdrefs

SoapInteger

SoapLanguage

SoapMonth

SoapMonthDay

SoapName

SoapNcName

SoapNegativeInteger

SoapNmtoken

SoapNmtokens

SoapNonNegativeInteger

SoapNonPositiveInteger

SoapNormalizedString

SoapNotation

SoapPositiveInteger

SoapQName

SoapTime

SoapToken

SoapYear

SoapYearMonth

Interfaces

ISoapXsd

ISoapXsd Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISoapXsd
```

Public Methods

```
System.String GetXsdType();
```

SoapAnyUri Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapAnyUri
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapAnyUri();
public SoapAnyUri(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapAnyUri Parse(
   System.String value
);
public System.String ToString();
```

SoapBase64Binary Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapBase64Binary
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapBase64Binary();
public SoapBase64Binary(
   ubyte[] value
);
```

Public Properties

```
public ubyte[] get_Value();
public void set_Value(
   ubyte[] value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapBase64Binary Parse(
   System.String value
);
public System.String ToString();
```

SoapDate Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapDate
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapDate();
public SoapDate(
   System.DateTime value
);
public SoapDate(
   System.DateTime value,
   int sign
);
```

Public Properties

```
public int get_Sign();
public void set_Sign(
   int value
);
public System.DateTime get_Value();
public void set_Value(
   System.DateTime value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapDate Parse(
   System.String value
);
public System.String ToString();
```

SoapDateTime Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapDateTime
   extends System.Object
```

Public Constructors

```
public SoapDateTime();
```

Public Properties

```
public static System.String get_XsdType();
```

Public Methods

```
public static System.DateTime Parse(
   System.String value
);
public static System.String ToString(
   System.DateTime value
);
```

SoapDay Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapDay
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapDay();
public SoapDay(
   System.DateTime value
);
```

Public Properties

```
public System.DateTime get_Value();
public void set_Value(
   System.DateTime value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapDay Parse(
   System.String value
);
public System.String ToString();
```

SoapDuration Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapDuration
   extends System.Object
```

Public Constructors

```
public SoapDuration();
```

Public Properties

```
public static System.String get_XsdType();
```

Public Methods

```
public static System.TimeSpan Parse(
   System.String value
);
public static System.String ToString(
   System.TimeSpan timeSpan
);
```

SoapEntities Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapEntities
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapEntities();
public SoapEntities(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapEntities Parse(
   System.String value
);
public System.String ToString();
```

SoapEntity Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapEntity
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapEntity();
public SoapEntity(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapEntity Parse(
   System.String value
);
public System.String ToString();
```

SoapHexBinary Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapHexBinary
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapHexBinary();
public SoapHexBinary(
   ubyte[] value
);
```

Public Properties

```
public ubyte[] get_Value();
public void set_Value(
   ubyte[] value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapHexBinary Parse(
   System.String value
);
public System.String ToString();
```

SoapId Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapId
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapId();
public SoapId(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapId Parse(
   System.String value
);
public System.String ToString();
```

SoapIdref Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapIdref
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapIdref();
public SoapIdref(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapIdref Parse(
   System.String value
);
public System.String ToString();
```

SoapIdrefs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapIdrefs
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapIdrefs();
public SoapIdrefs(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapIdrefs Parse(
   System.String value
);
public System.String ToString();
```

SoapInteger Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapInteger
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapInteger();
public SoapInteger(
   System.Decimal value
);
```

Public Properties

```
public System.Decimal get_Value();
public void set_Value(
   System.Decimal value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapInteger Parse(
   System.String value
);
public System.String ToString();
```

SoapLanguage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapLanguage
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapLanguage();
public SoapLanguage(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapLanguage Parse(
   System.String value
);
public System.String ToString();
```

SoapMonth Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapMonth
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapMonth();
public SoapMonth(
   System.DateTime value
);
```

Public Properties

```
public System.DateTime get_Value();
public void set_Value(
   System.DateTime value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapMonth Parse(
   System.String value
);
public System.String ToString();
```

SoapMonthDay Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapMonthDay
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapMonthDay();
public SoapMonthDay(
   System.DateTime value
);
```

Public Properties

```
public System.DateTime get_Value();
public void set_Value(
   System.DateTime value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapMonthDay Parse(
   System.String value
);
public System.String ToString();
```

SoapName Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapName
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapName();
public SoapName(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapName Parse(
   System.String value
);
public System.String ToString();
```

SoapNcName Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapNcName
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapNcName();
public SoapNcName(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapNcName Parse(
   System.String value
);
public System.String ToString();
```

SoapNegativeInteger Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapNegativeInteger
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapNegativeInteger();
public SoapNegativeInteger(
   System.Decimal value
);
```

Public Properties

```
public System.Decimal get_Value();
public void set_Value(
   System.Decimal value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapNegativeInteger Parse(
   System.String value
);
public System.String ToString();
```

SoapNmtoken Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapNmtoken
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapNmtoken();
public SoapNmtoken(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapNmtoken Parse(
   System.String value
);
public System.String ToString();
```

SoapNmtokens Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapNmtokens
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapNmtokens();
public SoapNmtokens(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapNmtokens Parse(
   System.String value
);
public System.String ToString();
```

SoapNonNegativeInteger Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapNonNegativeInteger
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapNonNegativeInteger();
public SoapNonNegativeInteger(
   System.Decimal value
);
```

Public Properties

```
public System.Decimal get_Value();
public void set_Value(
   System.Decimal value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapNonNegativeInteger Parse(
   System.String value
);
public System.String ToString();
```

SoapNonPositiveInteger Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapNonPositiveInteger
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapNonPositiveInteger();
public SoapNonPositiveInteger(
   System.Decimal value
);
```

Public Properties

```
public System.Decimal get_Value();
public void set_Value(
   System.Decimal value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapNonPositiveInteger Parse(
   System.String value
);
public System.String ToString();
```

SoapNormalizedString Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapNormalizedString
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapNormalizedString();
public SoapNormalizedString(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapNormalizedString Parse(
   System.String value
);
public System.String ToString();
```

SoapNotation Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapNotation
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapNotation();
public SoapNotation(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapNotation Parse(
   System.String value
);
public System.String ToString();
```

SoapPositiveInteger Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapPositiveInteger
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapPositiveInteger();
public SoapPositiveInteger(
   System.Decimal value
);
```

Public Properties

```
public System.Decimal get_Value();
public void set_Value(
   System.Decimal value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapPositiveInteger Parse(
   System.String value
);
public System.String ToString();
```

SoapQName Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapQName
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapQName();
public SoapQName(
   System.String value
);
public SoapQName(
   System.String key,
   System.String name
);
public SoapQName(
   System.String key,
   System.String name,
   System.String namespaceValue
);
```

Public Properties

```
public System.String get_Key();
public void set_Key(
   System.String value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapQName Parse(
   System.String value
);
public System.String ToString();
```

SoapTime Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapTime
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapTime();
public SoapTime(
   System.DateTime value
);
```

Public Properties

```
public System.DateTime get_Value();
public void set_Value(
   System.DateTime value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapTime Parse(
   System.String value
);
public System.String ToString();
```

SoapToken Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapToken
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapToken();
public SoapToken(
   System.String value
);
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapToken Parse(
   System.String value
);
public System.String ToString();
```

SoapYear Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapYear
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapYear();
public SoapYear(
   System.DateTime value
);
public SoapYear(
   System.DateTime value,
   int sign
);
```

Public Properties

```
public int get_Sign();
public void set_Sign(
   int value
);
public System.DateTime get_Value();
public void set_Value(
   System.DateTime value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapYear Parse(
   System.String value
);
public System.String ToString();
```

SoapYearMonth Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapYearMonth
   extends System.Object
   implements System.Runtime.Remoting.Metadata.W3cXsd2001.ISoapXsd
```

Public Constructors

```
public SoapYearMonth();
public SoapYearMonth(
   System.DateTime value
);
public SoapYearMonth(
   System.DateTime value,
   int sign
);
```

Public Properties

```
public int get_Sign();
public void set_Sign(
   int value
);
public System.DateTime get_Value();
public void set_Value(
   System.DateTime value
);
public static System.String get_XsdType();
```

Public Methods

```
public final System.String GetXsdType();
public static System.Runtime.Remoting.Metadata.W3cXsd2001.SoapYearMonth Parse(
   System.String value
);
public System.String ToString();
```

System.Runtime.Remoting.MetadataServices Namespace (Visual J# Syntax)

Classes

MetaData

SdlChannelSink

SdlChannelSinkProvider

ServiceType

SUDSGeneratorException

SUDSParserException

Enumerations

SdlType

MetaData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MetaData
   extends System.Object
```

Public Constructors

```
public MetaData();
```

Public Methods

```
public static void ConvertCodeSourceFileToAssemblyFile(
   System.String codePath,
   System.String assemblyPath,
   System.String strongNameFilename
);
public static void ConvertCodeSourceStreamToAssemblyFile(
   System.Collections.ArrayList outCodeStreamList,
   System.String assemblyPath,
   System.String strongNameFilename
);
public static void ConvertSchemaStreamToCodeSourceStream(
   boolean clientProxy,
   System.String outputDirectory,
   System.IO.Stream inputStream,
   System.Collections.ArrayList outCodeStreamList
);
public static void ConvertSchemaStreamToCodeSourceStream(
   boolean clientProxy,
   System.String outputDirectory,
   System.IO.Stream inputStream,
   System.Collections.ArrayList outCodeStreamList,
   System.String proxyUrl
);
public static void ConvertSchemaStreamToCodeSourceStream(
   boolean clientProxy,
   System.String outputDirectory,
   System.IO.Stream inputStream,
   System.Collections.ArrayList outCodeStreamList,
   System.String proxyUrl,
   System.String proxyNamespace
);
public static void ConvertTypesToSchemaToFile(
   System.Runtime.Remoting.MetadataServices.ServiceType[] types,
   System.Runtime.Remoting.MetadataServices.SdlType sdlType,
   System.String path
);
public static void ConvertTypesToSchemaToFile(
   System.Type[] types,
   System.Runtime.Remoting.MetadataServices.SdlType sdlType,
   System.String path
);
public static void ConvertTypesToSchemaToStream(
   System.Runtime.Remoting.MetadataServices.ServiceType[] serviceTypes,
   System.Runtime.Remoting.MetadataServices.SdlType sdlType,
   System.IO.Stream outputStream
);
```

```
public static void ConvertTypesToSchemaToStream(
   System.Type[] types,
   System.Runtime.Remoting.MetadataServices.SdlType sdlType,
   System.IO.Stream outputStream
);
public static void RetrieveSchemaFromUrlToFile(
   System.String url,
   System.String path
);
public static void RetrieveSchemaFromUrlToStream(
   System.String url,
   System.IO.Stream outputStream
);
public static void SaveStreamToFile(
   System.IO.Stream inputStream,
   System.String path
);
```

SdlChannelSink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SdlChannelSink
   extends System.Object
   implements System.Runtime.Remoting.Channels.IServerChannelSink, System.Runtime.Remoting.Ch
annels.IChannelSinkBase
```

Public Constructors

```
public SdlChannelSink(
   System.Runtime.Remoting.Channels.IChannelReceiver receiver,
   System.Runtime.Remoting.Channels.IServerChannelSink nextSink
);
```

Public Properties

```
public final System.Runtime.Remoting.Channels.IServerChannelSink get_NextChannelSink();
public final System.Collections.IDictionary get_Properties();
```

Public Methods

```
public final void AsyncProcessResponse(
   System.Runtime.Remoting.Channels.IServerResponseChannelSinkStack sinkStack,
   System.Object state,
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers,
   System.IO.Stream stream
);
public final System.IO.Stream GetResponseStream(
   System.Runtime.Remoting.Channels.IServerResponseChannelSinkStack sinkStack,
   System.Object state,
   System.Runtime.Remoting.Messaging.IMessage msg,
   System.Runtime.Remoting.Channels.ITransportHeaders headers
);
public final System.Runtime.Remoting.Channels.ServerProcessing ProcessMessage(
   System.Runtime.Remoting.Channels.IServerChannelSinkStack sinkStack,
   System.Runtime.Remoting.Messaging.IMessage requestMsg,
   System.Runtime.Remoting.Channels.ITransportHeaders requestHeaders,
   System.IO.Stream requestStream,
   System.Runtime.Remoting.Messaging.IMessage& responseMsg,
   System.Runtime.Remoting.Channels.ITransportHeaders& responseHeaders,
   System.IO.Stream& responseStream
);
```

SdlChannelSinkProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SdlChannelSinkProvider
   extends System.Object
   implements System.Runtime.Remoting.Channels.IServerChannelSinkProvider
```

Public Constructors

```
public SdlChannelSinkProvider();
public SdlChannelSinkProvider(
   System.Collections.IDictionary properties,
   System.Collections.ICollection providerData
);
```

Public Properties

```
public final System.Runtime.Remoting.Channels.IServerChannelSinkProvider get_Next();
public final void set_Next(
   System.Runtime.Remoting.Channels.IServerChannelSinkProvider value
);
```

Public Methods

```
public final System.Runtime.Remoting.Channels.IServerChannelSink CreateSink(
   System.Runtime.Remoting.Channels.IChannelReceiver channel
);
public final void GetChannelData(
   System.Runtime.Remoting.Channels.IChannelDataStore localChannelData
);
```

SdlType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SdlType
```

Members

Sdl

Wsdl

ServiceType Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ServiceType
   extends System.Object
```

Public Constructors

```
public ServiceType(
   System.Type type
);
public ServiceType(
   System.Type type,
   System.String url
);
```

Public Properties

```
public System.Type get_ObjectType();
public System.String get_Url();
```

SUDSGeneratorException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SUDSGeneratorException
   extends System.Exception
```

SUDSParserException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SUDSParserException
   extends System.Exception
```

System.Runtime.Remoting.Proxies Namespace (Visual J# Syntax)

Classes

ProxyAttribute

RealProxy

ProxyAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class ProxyAttribute
   extends System.Attribute
   implements System.Runtime.Remoting.Contexts.IContextAttribute
```

Public Constructors

```
public ProxyAttribute();
```

Public Methods

```
public System.MarshalByRefObject CreateInstance(
   System.Type serverType
);
public System.Runtime.Remoting.Proxies.RealProxy CreateProxy(
   System.Runtime.Remoting.ObjRef objRef,
   System.Type serverType,
   System.Object serverObject,
   System.Runtime.Remoting.Contexts.Context serverContext
);
public final void GetPropertiesForNewContext(
   System.Runtime.Remoting.Activation.IConstructionCallMessage msg
);
public final boolean IsContextOK(
   System.Runtime.Remoting.Contexts.Context ctx,
   System.Runtime.Remoting.Activation.IConstructionCallMessage msg
);
```

RealProxy Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class RealProxy
   extends System.Object
```

Public Methods

```
public System.Runtime.Remoting.ObjRef CreateObjRef(
   System.Type requestedType
);
public System.IntPtr GetCOMIUnknown(
   boolean fIsMarshalled
);
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public System.Type GetProxiedType();
public static System.Object GetStubData(
   System.Runtime.Remoting.Proxies.RealProxy rp
);
public System.Object GetTransparentProxy();
public System.Runtime.Remoting.Activation.IConstructionReturnMessage InitializeServerObject(
   System.Runtime.Remoting.Activation.IConstructionCallMessage ctorMsg
);
public abstract System.Runtime.Remoting.Messaging.IMessage Invoke(
   System.Runtime.Remoting.Messaging.IMessage msg
);
public void SetCOMIUnknown(
   System.IntPtr i
);
public static void SetStubData(
   System.Runtime.Remoting.Proxies.RealProxy rp,
   System.Object stubData
);
public System.IntPtr SupportsInterface(
   System.Guid& iid
);
```

Protected Constructors

```
protected RealProxy();
protected RealProxy(
   System.Type classToProxy
);
protected RealProxy(
   System.Type classToProxy,
   System.IntPtr stub,
   System.Object stubData
);
```

Protected Methods

```
protected void AttachServer(
   System.MarshalByRefObject s
```

```
);
protected System.MarshalByRefObject DetachServer();
protected System.MarshalByRefObject GetUnwrappedServer();
```

System.Runtime.Remoting.Services Namespace (Visual J# Syntax)

Classes

EnterpriseServicesHelper

RemotingClientProxy

RemotingService

TrackingServices

Interfaces

ITrackingHandler

EnterpriseServicesHelper Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EnterpriseServicesHelper
   extends System.Object
```

Public Constructors

```
public EnterpriseServicesHelper();
```

Public Methods

```
public static System.Runtime.Remoting.Activation.IConstructionReturnMessage CreateConstructio
nReturnMessage(
   System.Runtime.Remoting.Activation.IConstructionCallMessage ctorMsg,
   System.MarshalByRefObject retObj
);
public static void SwitchWrappers(
   System.Runtime.Remoting.Proxies.RealProxy oldcp,
   System.Runtime.Remoting.Proxies.RealProxy newcp
);
public static System.Object WrapIUnknownWithComObject(
   System.IntPtr punk
);
```

ITrackingHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ITrackingHandler
```

Public Methods

```
void DisconnectedObject(
   System.Object obj
);
void MarshaledObject(
   System.Object obj,
   System.Runtime.Remoting.ObjRef or
);
void UnmarshaledObject(
   System.Object obj,
   System.Runtime.Remoting.ObjRef or
);
```

RemotingClientProxy Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class RemotingClientProxy
   extends System.ComponentModel.Component
```

Public Properties

```
public boolean get_AllowAutoRedirect();
public void set_AllowAutoRedirect(
   boolean value
);
public System.Object get_Cookies();
public System.String get_Domain();
public void set_Domain(
   System.String value
);
public boolean get_EnableCookies();
public void set_EnableCookies(
   boolean value
);
public System.String get_Password();
public void set_Password(
   System.String value
);
public System.String get_Path();
public void set_Path(
   System.String value
);
public boolean get_PreAuthenticate();
public void set_PreAuthenticate(
   boolean value
);
public System.String get_ProxyName();
public void set_ProxyName(
   System.String value
);
public int get_ProxyPort();
public void set_ProxyPort(
   int value
);
public int get_Timeout();
public void set_Timeout(
   int value
);
public System.String get_Url();
public void set_Url(
   System.String value
);
public System.String get_UserAgent();
public void set_UserAgent(
   System.String value
);
public System.String get_Username();
public void set_Username(
   System.String value
);
```

Protected Constructors

```
protected RemotingClientProxy();
```

Protected Methods

```
protected void ConfigureProxy(
   System.Type type,
   System.String url
);
protected void ConnectProxy();
```

Protected Fields

```
protected System.Object _tp;
protected System.Type _type;
protected System.String _url;
```

RemotingService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RemotingService
   extends System.ComponentModel.Component
```

Public Constructors

```
public RemotingService();
```

Public Properties

```
public System.Web.HttpApplicationState get_Application();
public System.Web.HttpContext get_Context();
public System.Web.HttpServerUtility get_Server();
public System.Web.SessionState.HttpSessionState get_Session();
public System.Security.Principal.IPrincipal get_User();
```

TrackingServices Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TrackingServices
   extends System.Object
```

Public Constructors

```
public TrackingServices();
```

Public Properties

```
public static System.Runtime.Remoting.Services.ITrackingHandler[] get_RegisteredHandlers();
```

Public Methods

```
public static void RegisterTrackingHandler(
   System.Runtime.Remoting.Services.ITrackingHandler handler
);
public static void UnregisterTrackingHandler(
   System.Runtime.Remoting.Services.ITrackingHandler handler
);
```

System.Runtime.Serialization Namespace (Visual J# Syntax)

Classes

Formatter

FormatterConverter

FormatterServices

ObjectIDGenerator

ObjectManager

SerializationBinder

SerializationException

SerializationInfo

SerializationInfoEnumerator

SurrogateSelector

Interfaces

IDeserializationCallback

IFormatter

IFormatterConverter

IObjectReference

ISerializable

ISerializationSurrogate

ISurrogateSelector

Structures

SerializationEntry

StreamingContext

Enumerations

StreamingContextStates

Formatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public abstract class Formatter
   extends System.Object
   implements System.Runtime.Serialization.IFormatter
```

Public Properties

```
public abstract System.Runtime.Serialization.SerializationBinder get_Binder();
public abstract void set_Binder(
   System.Runtime.Serialization.SerializationBinder value
);
public abstract System.Runtime.Serialization.StreamingContext get_Context();
public abstract void set_Context(
   System.Runtime.Serialization.StreamingContext value
);
public abstract System.Runtime.Serialization.ISurrogateSelector get_SurrogateSelector();
public abstract void set_SurrogateSelector(
   System.Runtime.Serialization.ISurrogateSelector value
);
```

Public Methods

```
public abstract System.Object Deserialize(
   System.IO.Stream serializationStream
);
public abstract void Serialize(
   System.IO.Stream serializationStream,
   System.Object graph
);
```

Protected Constructors

```
protected Formatter();
```

Protected Methods

```
protected System.Object GetNext(
   long objID
);
protected long Schedule(
   System.Object obj
);
protected abstract void WriteArray(
   System.Object obj,
   System.String name,
   System.Type memberType
);
protected abstract void WriteBoolean(
   boolean val,
   System.String name
```

```
);
protected abstract void WriteByte(
   ubyte val,
   System.String name
);
protected abstract void WriteChar(
   char val,
   System.String name
);
protected abstract void WriteDateTime(
   System.DateTime val,
   System.String name
);
protected abstract void WriteDecimal(
   System.Decimal val,
   System.String name
);
protected abstract void WriteDouble(
   double val,
   System.String name
);
protected abstract void WriteInt16(
   short val,
   System.String name
);
protected abstract void WriteInt32(
   int val,
   System.String name
);
protected abstract void WriteInt64(
   long val,
   System.String name
);
protected void WriteMember(
   System.String memberName,
   System.Object data
);
protected abstract void WriteObjectRef(
   System.Object obj,
   System.String name,
   System.Type memberType
);
protected abstract void WriteSingle(
   float val,
   System.String name
);
protected abstract void WriteTimeSpan(
   System.TimeSpan val,
   System.String name
);
protected abstract void WriteValueType(
   System.Object obj,
   System.String name,
   System.Type memberType
);
```

Protected Fields

```
protected System.Runtime.Serialization.ObjectIDGenerator m_idGenerator;
protected System.Collections.Queue m_objectQueue;
```

FormatterConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FormatterConverter
   extends System.Object
   implements System.Runtime.Serialization.IFormatterConverter
```

Public Constructors

```
public FormatterConverter();
```

Public Methods

```
public final System.Object Convert(
   System.Object value,
   System.Type type
);
public final System.Object Convert(
   System.Object value,
   System.TypeCode typeCode
);
public final boolean ToBoolean(
   System.Object value
);
public final ubyte ToByte(
   System.Object value
);
public final char ToChar(
   System.Object value
);
public final System.DateTime ToDateTime(
   System.Object value
);
public final System.Decimal ToDecimal(
   System.Object value
);
public final double ToDouble(
   System.Object value
);
public final short ToInt16(
   System.Object value
);
public final int ToInt32(
   System.Object value
);
public final long ToInt64(
   System.Object value
);
public final float ToSingle(
   System.Object value
);
public final System.String ToString(
   System.Object value
);
```

FormatterServices Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FormatterServices
   extends System.Object
```

Public Methods

```
public static System.Object[] GetObjectData(
   System.Object obj,
   System.Reflection.MemberInfo[] members
);
public static System.Reflection.MemberInfo[] GetSerializableMembers(
   System.Type type
);
public static System.Reflection.MemberInfo[] GetSerializableMembers(
   System.Type type,
   System.Runtime.Serialization.StreamingContext context
);
public static System.Type GetTypeFromAssembly(
   System.Reflection.Assembly assem,
   System.String name
);
public static System.Object GetUninitializedObject(
   System.Type type
);
public static System.Object PopulateObjectMembers(
   System.Object obj,
   System.Reflection.MemberInfo[] members,
   System.Object[] data
);
```

IDeserializationCallback Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDeserializationCallback
```

Public Methods

```
void OnDeserialization(
   System.Object sender
);
```

IFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IFormatter
```

Public Properties

```
System.Runtime.Serialization.SerializationBinder get_Binder();
void set_Binder(
   System.Runtime.Serialization.SerializationBinder value
);
System.Runtime.Serialization.StreamingContext get_Context();
void set_Context(
   System.Runtime.Serialization.StreamingContext value
);
System.Runtime.Serialization.ISurrogateSelector get_SurrogateSelector();
void set_SurrogateSelector(
   System.Runtime.Serialization.ISurrogateSelector value
);
```

Public Methods

```
System.Object Deserialize(
   System.IO.Stream serializationStream
);
void Serialize(
   System.IO.Stream serializationStream,
   System.Object graph
);
```

IFormatterConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.CLSCompliantAttribute(IsCompliant = false) */
public interface IFormatterConverter
```

Public Methods

```
System.Object Convert(
   System.Object value,
   System.Type type
);
System.Object Convert(
   System.Object value,
   System.TypeCode typeCode
);
boolean ToBoolean(
   System.Object value
);
ubyte ToByte(
   System.Object value
);
char ToChar(
   System.Object value
);
System.DateTime ToDateTime(
   System.Object value
);
System.Decimal ToDecimal(
   System.Object value
);
double ToDouble(
   System.Object value
);
short ToInt16(
   System.Object value
);
int ToInt32(
   System.Object value
);
long ToInt64(
   System.Object value
);
byte ToSByte(
   System.Object value
);
float ToSingle(
   System.Object value
);
System.String ToString(
   System.Object value
);
System.UInt16 ToUInt16(
   System.Object value
);
System.UInt32 ToUInt32(
   System.Object value
);
System.UInt64 ToUInt64(
   System.Object value
```

```
);
```

IObjectReference Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IObjectReference
```

Public Methods

```
System.Object GetRealObject(
   System.Runtime.Serialization.StreamingContext context
);
```

ISerializable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISerializable
```

Public Methods

```
void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ISerializationSurrogate Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISerializationSurrogate
```

Public Methods

```
void GetObjectData(
   System.Object obj,
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
System.Object SetObjectData(
   System.Object obj,
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context,
   System.Runtime.Serialization.ISurrogateSelector selector
);
```

ISurrogateSelector Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISurrogateSelector
```

Public Methods

```
void ChainSelector(
   System.Runtime.Serialization.ISurrogateSelector selector
);
System.Runtime.Serialization.ISurrogateSelector GetNextSelector();
System.Runtime.Serialization.ISerializationSurrogate GetSurrogate(
   System.Type type,
   System.Runtime.Serialization.StreamingContext context,
   System.Runtime.Serialization.ISurrogateSelector& selector
);
```

ObjectIDGenerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ObjectIDGenerator
   extends System.Object
```

Public Constructors

```
public ObjectIDGenerator();
```

Public Methods

```
public long GetId(
   System.Object obj,
   boolean firstTime
);
public long HasId(
   System.Object obj,
   boolean firstTime
);
```

ObjectManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ObjectManager
   extends System.Object
```

Public Constructors

```
public ObjectManager(
   System.Runtime.Serialization.ISurrogateSelector selector,
   System.Runtime.Serialization.StreamingContext context
);
```

Public Methods

```
public void DoFixups();
public System.Object GetObject(
   long objectID
);
public void RaiseDeserializationEvent();
public void RecordArrayElementFixup(
   long arrayToBeFixed,
   int index,
   long objectRequired
);
public void RecordArrayElementFixup(
   long arrayToBeFixed,
   int[] indices,
   long objectRequired
);
public void RecordDelayedFixup(
   long objectToBeFixed,
   System.String memberName,
   long objectRequired
);
public void RecordFixup(
   long objectToBeFixed,
   System.Reflection.MemberInfo member,
   long objectRequired
);
public void RegisterObject(
   System.Object obj,
   long objectID
);
public void RegisterObject(
   System.Object obj,
   long objectID,
   System.Runtime.Serialization.SerializationInfo info
);
public void RegisterObject(
   System.Object obj,
   long objectID,
   System.Runtime.Serialization.SerializationInfo info,
   long idOfContainingObj,
   System.Reflection.MemberInfo member
);
public void RegisterObject(
   System.Object obj,
```

```
    long objectID,
    System.Runtime.Serialization.SerializationInfo info,
    long idOfContainingObj,
    System.Reflection.MemberInfo member,
    int[] arrayIndex
);
```

SerializationBinder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class SerializationBinder
   extends System.Object
```

Public Methods

```
public abstract System.Type BindToType(
   System.String assemblyName,
   System.String typeName
);
```

Protected Constructors

```
protected SerializationBinder();
```

SerializationEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SerializationEntry
   extends System.ValueType
```

Public Properties

```
public System.String get_Name();
public System.Type get_ObjectType();
public System.Object get_Value();
```

SerializationException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SerializationException
   extends System.SystemException
```

Public Constructors

```
public SerializationException();
public SerializationException(
   System.String message
);
public SerializationException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected SerializationException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

SerializationInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SerializationInfo
   extends System.Object
```

Public Properties

```
public System.String get_AssemblyName();
public void set_AssemblyName(
   System.String value
);
public System.String get_FullTypeName();
public void set_FullTypeName(
   System.String value
);
public int get_MemberCount();
```

Public Methods

```
public void AddValue(
   System.String name,
   boolean value
);
public void AddValue(
   System.String name,
   ubyte value
);
public void AddValue(
   System.String name,
   char value
);
public void AddValue(
   System.String name,
   double value
);
public void AddValue(
   System.String name,
   short value
);
public void AddValue(
   System.String name,
   int value
);
public void AddValue(
   System.String name,
   long value
);
public void AddValue(
   System.String name,
   float value
);
public void AddValue(
   System.String name,
   System.DateTime value
);
public void AddValue(
   System.String name,
```

```
   System.Decimal value
);
public void AddValue(
   System.String name,
   System.Object value
);
public void AddValue(
   System.String name,
   System.Object value,
   System.Type type
);
public boolean GetBoolean(
   System.String name
);
public ubyte GetByte(
   System.String name
);
public char GetChar(
   System.String name
);
public System.DateTime GetDateTime(
   System.String name
);
public System.Decimal GetDecimal(
   System.String name
);
public double GetDouble(
   System.String name
);
public System.Runtime.Serialization.SerializationInfoEnumerator GetEnumerator();
public short GetInt16(
   System.String name
);
public int GetInt32(
   System.String name
);
public long GetInt64(
   System.String name
);
public float GetSingle(
   System.String name
);
public System.String GetString(
   System.String name
);
public System.Object GetValue(
   System.String name,
   System.Type type
);
public void SetType(
   System.Type type
);
```

SerializationInfoEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SerializationInfoEnumerator
   extends System.Object
   implements System.Collections.IEnumerator
```

Public Properties

```
public System.Runtime.Serialization.SerializationEntry get_Current();
public System.String get_Name();
public System.Type get_ObjectType();
public System.Object get_Value();
```

Public Methods

```
public final boolean MoveNext();
public final void Reset();
```

StreamingContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StreamingContext
   extends System.ValueType
```

Public Constructors

```
public StreamingContext(
   System.Runtime.Serialization.StreamingContextStates state
);
public StreamingContext(
   System.Runtime.Serialization.StreamingContextStates state,
   System.Object additional
);
```

Public Properties

```
public System.Object get_Context();
public System.Runtime.Serialization.StreamingContextStates get_State();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

StreamingContextStates Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum StreamingContextStates
```

Members

All

Clone

CrossAppDomain

CrossMachine

CrossProcess

File

Other

Persistence

Remoting

SurrogateSelector Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SurrogateSelector
   extends System.Object
   implements System.Runtime.Serialization.ISurrogateSelector
```

Public Constructors

```
public SurrogateSelector();
```

Public Methods

```
public void AddSurrogate(
   System.Type type,
   System.Runtime.Serialization.StreamingContext context,
   System.Runtime.Serialization.ISerializationSurrogate surrogate
);
public void ChainSelector(
   System.Runtime.Serialization.ISurrogateSelector selector
);
public System.Runtime.Serialization.ISurrogateSelector GetNextSelector();
public System.Runtime.Serialization.ISerializationSurrogate GetSurrogate(
   System.Type type,
   System.Runtime.Serialization.StreamingContext context,
   System.Runtime.Serialization.ISurrogateSelector& selector
);
public void RemoveSurrogate(
   System.Type type,
   System.Runtime.Serialization.StreamingContext context
);
```

System.Runtime.Serialization.Formatters Namespace (Visual J# Syntax)

Classes

InternalRM

InternalST

ServerFault

SoapFault

SoapMessage

Interfaces

IFieldInfo

ISoapMessage

Enumerations

FormatterAssemblyStyle

FormatterTypeStyle

FormatterAssemblyStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum FormatterAssemblyStyle
```

Members

Full

Simple

FormatterTypeStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum FormatterTypeStyle
```

Members

TypesAlways

TypesWhenNeeded

XsdString

IFieldInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IFieldInfo
```

Public Properties

```
System.String[] get_FieldNames();
void set_FieldNames(
   System.String[] value
);
System.Type[] get_FieldTypes();
void set_FieldTypes(
   System.Type[] value
);
```

InternalRM Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InternalRM
   extends System.Object
```

InternalST Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InternalST
   extends System.Object
```

ISoapMessage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISoapMessage
```

Public Properties

```
System.Runtime.Remoting.Messaging.Header[] get_Headers();
void set_Headers(
   System.Runtime.Remoting.Messaging.Header[] value
);
System.String get_MethodName();
void set_MethodName(
   System.String value
);
System.String[] get_ParamNames();
void set_ParamNames(
   System.String[] value
);
System.Type[] get_ParamTypes();
void set_ParamTypes(
   System.Type[] value
);
System.Object[] get_ParamValues();
void set_ParamValues(
   System.Object[] value
);
System.String get_XmlNameSpace();
void set_XmlNameSpace(
   System.String value
);
```

ServerFault Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.Remoting.Metadata.SoapTypeAttribute(SoapOptions = None, XmlFiel
dOrder = All, UseAttribute = false, Embedded = true) */
public class ServerFault
   extends System.Object
```

Public Constructors

```
public ServerFault(
   System.String exceptionType,
   System.String message,
   System.String stackTrace
);
```

Public Properties

```
public System.String get_ExceptionMessage();
public void set_ExceptionMessage(
   System.String value
);
public System.String get_ExceptionType();
public void set_ExceptionType(
   System.String value
);
public System.String get_StackTrace();
public void set_StackTrace(
   System.String value
);
```

SoapFault Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.Remoting.Metadata.SoapTypeAttribute(SoapOptions = None, XmlFiel
dOrder = All, UseAttribute = false, Embedded = true) */
public class SoapFault
   extends System.Object
   implements System.Runtime.Serialization.ISerializable
```

Public Constructors

```
public SoapFault();
public SoapFault(
   System.String faultCode,
   System.String faultString,
   System.String faultActor,
   System.Runtime.Serialization.Formatters.ServerFault serverFault
);
```

Public Properties

```
public System.Object get_Detail();
public void set_Detail(
   System.Object value
);
public System.String get_FaultActor();
public void set_FaultActor(
   System.String value
);
public System.String get_FaultCode();
public void set_FaultCode(
   System.String value
);
public System.String get_FaultString();
public void set_FaultString(
   System.String value
);
```

Public Methods

```
public final void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

SoapMessage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SoapMessage
   extends System.Object
   implements System.Runtime.Serialization.Formatters.ISoapMessage
```

Public Constructors

```
public SoapMessage();
```

Public Properties

```
public final System.Runtime.Remoting.Messaging.Header[] get_Headers();
public final void set_Headers(
   System.Runtime.Remoting.Messaging.Header[] value
);
public final System.String get_MethodName();
public final void set_MethodName(
   System.String value
);
public final System.String[] get_ParamNames();
public final void set_ParamNames(
   System.String[] value
);
public final System.Type[] get_ParamTypes();
public final void set_ParamTypes(
   System.Type[] value
);
public final System.Object[] get_ParamValues();
public final void set_ParamValues(
   System.Object[] value
);
public final System.String get_XmlNameSpace();
public final void set_XmlNameSpace(
   System.String value
);
```

System.Runtime.Serialization.Formatters.Binary Namespace (Visual J# Syntax)

Classes

BinaryFormatter

BinaryFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BinaryFormatter
   extends System.Object
   implements System.Runtime.Remoting.Messaging.IRemotingFormatter, System.Runtime.Serializat
ion.IFormatter
```

Public Constructors

```
public BinaryFormatter();
public BinaryFormatter(
   System.Runtime.Serialization.ISurrogateSelector selector,
   System.Runtime.Serialization.StreamingContext context
);
```

Public Properties

```
public System.Runtime.Serialization.Formatters.FormatterAssemblyStyle get_AssemblyFormat();
public void set_AssemblyFormat(
   System.Runtime.Serialization.Formatters.FormatterAssemblyStyle value
);
public final System.Runtime.Serialization.SerializationBinder get_Binder();
public final void set_Binder(
   System.Runtime.Serialization.SerializationBinder value
);
public final System.Runtime.Serialization.StreamingContext get_Context();
public final void set_Context(
   System.Runtime.Serialization.StreamingContext value
);
public final System.Runtime.Serialization.ISurrogateSelector get_SurrogateSelector();
public final void set_SurrogateSelector(
   System.Runtime.Serialization.ISurrogateSelector value
);
public System.Runtime.Serialization.Formatters.FormatterTypeStyle get_TypeFormat();
public void set_TypeFormat(
   System.Runtime.Serialization.Formatters.FormatterTypeStyle value
);
```

Public Methods

```
public final System.Object Deserialize(
   System.IO.Stream serializationStream
);
public final System.Object Deserialize(
   System.IO.Stream serializationStream,
   System.Runtime.Remoting.Messaging.HeaderHandler handler
);
public System.Object DeserializeMethodResponse(
   System.IO.Stream serializationStream,
   System.Runtime.Remoting.Messaging.HeaderHandler handler,
   System.Runtime.Remoting.Messaging.IMethodCallMessage methodCallMessage
);
public final void Serialize(
   System.IO.Stream serializationStream,
   System.Object graph
```

```
);
public final void Serialize(
   System.IO.Stream serializationStream,
   System.Object graph,
   System.Runtime.Remoting.Messaging.Header[] headers
);
```

System.Runtime.Serialization.Formatters.Soap Namespace (Visual J# Syntax)

Classes

SoapFormatter

SoapFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapFormatter
   extends System.Object
   implements System.Runtime.Remoting.Messaging.IRemotingFormatter, System.Runtime.Serializat
ion.IFormatter
```

Public Constructors

```
public SoapFormatter();
public SoapFormatter(
   System.Runtime.Serialization.ISurrogateSelector selector,
   System.Runtime.Serialization.StreamingContext context
);
```

Public Properties

```
public System.Runtime.Serialization.Formatters.FormatterAssemblyStyle get_AssemblyFormat();
public void set_AssemblyFormat(
   System.Runtime.Serialization.Formatters.FormatterAssemblyStyle value
);
public final System.Runtime.Serialization.SerializationBinder get_Binder();
public final void set_Binder(
   System.Runtime.Serialization.SerializationBinder value
);
public final System.Runtime.Serialization.StreamingContext get_Context();
public final void set_Context(
   System.Runtime.Serialization.StreamingContext value
);
public final System.Runtime.Serialization.ISurrogateSelector get_SurrogateSelector();
public final void set_SurrogateSelector(
   System.Runtime.Serialization.ISurrogateSelector value
);
public System.Runtime.Serialization.Formatters.ISoapMessage get_TopObject();
public void set_TopObject(
   System.Runtime.Serialization.Formatters.ISoapMessage value
);
public System.Runtime.Serialization.Formatters.FormatterTypeStyle get_TypeFormat();
public void set_TypeFormat(
   System.Runtime.Serialization.Formatters.FormatterTypeStyle value
);
```

Public Methods

```
public final System.Object Deserialize(
   System.IO.Stream serializationStream
);
public final System.Object Deserialize(
   System.IO.Stream serializationStream,
   System.Runtime.Remoting.Messaging.HeaderHandler handler
);
public final void Serialize(
   System.IO.Stream serializationStream,
   System.Object graph
);
```

```
public final void Serialize(
   System.IO.Stream serializationStream,
   System.Object graph,
   System.Runtime.Remoting.Messaging.Header[] headers
);
```

System.Security Namespace (Visual J# Syntax)

Classes

AllowPartiallyTrustedCallersAttribute

CodeAccessPermission

NamedPermissionSet

PermissionSet

SecurityElement

SecurityException

SecurityManager

SuppressUnmanagedCodeSecurityAttribute

UnverifiableCodeAttribute

VerificationException

XmlSyntaxException

Interfaces

IEvidenceFactory

IPermission

ISecurityEncodable

ISecurityPolicyEncodable

IStackWalk

Enumerations

PolicyLevelType

SecurityZone

AllowPartiallyTrustedCallersAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = false) */
public class AllowPartiallyTrustedCallersAttribute
   extends System.Attribute
```

Public Constructors

```
public AllowPartiallyTrustedCallersAttribute();
```

CodeAccessPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class CodeAccessPermission
   extends System.Object
   implements System.Security.IPermission, System.Security.ISecurityEncodable, System.Securit
y.IStackWalk
```

Public Methods

```
public final void Assert();
public abstract System.Security.IPermission Copy();
public final void Demand();
public final void Deny();
public abstract void FromXml(
   System.Security.SecurityElement elem
);
public abstract System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public abstract boolean IsSubsetOf(
   System.Security.IPermission target
);
public final void PermitOnly();
public static void RevertAll();
public static void RevertAssert();
public static void RevertDeny();
public static void RevertPermitOnly();
public System.String ToString();
public abstract System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission other
);
```

Protected Constructors

```
protected CodeAccessPermission();
```

IEvidenceFactory Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IEvidenceFactory
```

Public Properties

```
System.Security.Policy.Evidence get_Evidence();
```

IPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IPermission
   implements System.Security.ISecurityEncodable
```

Public Methods

```
System.Security.IPermission Copy();
void Demand();
System.Security.IPermission Intersect(
   System.Security.IPermission target
);
boolean IsSubsetOf(
   System.Security.IPermission target
);
System.Security.IPermission Union(
   System.Security.IPermission target
);
```

ISecurityEncodable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISecurityEncodable
```

Public Methods

```
void FromXml(
   System.Security.SecurityElement e
);
System.Security.SecurityElement ToXml();
```

ISecurityPolicyEncodable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ISecurityPolicyEncodable
```

Public Methods

```
void FromXml(
   System.Security.SecurityElement e,
   System.Security.Policy.PolicyLevel level
);
System.Security.SecurityElement ToXml(
   System.Security.Policy.PolicyLevel level
);
```

IStackWalk Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IStackWalk
```

Public Methods

```
void Assert();
void Demand();
void Deny();
void PermitOnly();
```

NamedPermissionSet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class NamedPermissionSet
   extends System.Security.PermissionSet
```

Public Constructors

```
public NamedPermissionSet(
   System.Security.NamedPermissionSet permSet
);
public NamedPermissionSet(
   System.String name
);
public NamedPermissionSet(
   System.String name,
   System.Security.PermissionSet permSet
);
public NamedPermissionSet(
   System.String name,
   System.Security.Permissions.PermissionState state
);
```

Public Properties

```
public System.String get_Description();
public void set_Description(
   System.String value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
```

Public Methods

```
public System.Security.PermissionSet Copy();
public System.Security.NamedPermissionSet Copy(
   System.String name
);
public void FromXml(
   System.Security.SecurityElement et
);
public System.Security.SecurityElement ToXml();
```

PermissionSet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class PermissionSet
   extends System.Object
   implements System.Security.ISecurityEncodable, System.Collections.ICollection, System.Coll
ections.IEnumerable, System.Security.IStackWalk, System.Runtime.Serialization.IDeserializatio
nCallback
```

Public Constructors

```
public PermissionSet(
   System.Security.PermissionSet permSet
);
public PermissionSet(
   System.Security.Permissions.PermissionState state
);
```

Public Properties

```
public int get_Count();
public boolean get_IsReadOnly();
public boolean get_IsSynchronized();
public System.Object get_SyncRoot();
```

Public Methods

```
public System.Security.IPermission AddPermission(
   System.Security.IPermission perm
);
public void Assert();
public boolean ContainsNonCodeAccessPermissions();
public static ubyte[] ConvertPermissionSet(
   System.String inFormat,
   ubyte[] inData,
   System.String outFormat
);
public System.Security.PermissionSet Copy();
public void CopyTo(
   System.Array array,
   int index
);
public void Demand();
public void Deny();
public void FromXml(
   System.Security.SecurityElement et
);
public System.Collections.IEnumerator GetEnumerator();
public System.Security.IPermission GetPermission(
   System.Type permClass
);
public System.Security.PermissionSet Intersect(
   System.Security.PermissionSet other
);
public boolean IsEmpty();
```

```
public boolean IsSubsetOf(
   System.Security.PermissionSet target
);
public boolean IsUnrestricted();
public void PermitOnly();
public System.Security.IPermission RemovePermission(
   System.Type permClass
);
public System.Security.IPermission SetPermission(
   System.Security.IPermission perm
);
public System.String ToString();
public System.Security.SecurityElement ToXml();
public System.Security.PermissionSet Union(
   System.Security.PermissionSet other
);
```

PolicyLevelType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PolicyLevelType
```

Members

AppDomain

Enterprise

Machine

User

SecurityElement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SecurityElement
   extends System.Object
```

Public Constructors

```
public SecurityElement(
   System.String tag
);
public SecurityElement(
   System.String tag,
   System.String text
);
```

Public Properties

```
public System.Collections.Hashtable get_Attributes();
public void set_Attributes(
   System.Collections.Hashtable value
);
public System.Collections.ArrayList get_Children();
public void set_Children(
   System.Collections.ArrayList value
);
public System.String get_Tag();
public void set_Tag(
   System.String value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Methods

```
public void AddAttribute(
   System.String name,
   System.String value
);
public void AddChild(
   System.Security.SecurityElement child
);
public System.String Attribute(
   System.String name
);
public boolean Equal(
   System.Security.SecurityElement other
);
public static System.String Escape(
   System.String str
);
public static boolean IsValidAttributeName(
   System.String name
```

```
);
public static boolean IsValidAttributeValue(
   System.String value
);
public static boolean IsValidTag(
   System.String tag
);
public static boolean IsValidText(
   System.String text
);
public System.Security.SecurityElement SearchForChildByTag(
   System.String tag
);
public System.String SearchForTextOfTag(
   System.String tag
);
public System.String ToString();
```

SecurityException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SecurityException
   extends System.SystemException
```

Public Constructors

```
public SecurityException();
public SecurityException(
   System.String message
);
public SecurityException(
   System.String message,
   System.Exception inner
);
public SecurityException(
   System.String message,
   System.Type type
);
public SecurityException(
   System.String message,
   System.Type type,
   System.String state
);
```

Public Properties

```
public System.String get_PermissionState();
public System.Type get_PermissionType();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public System.String ToString();
```

Protected Constructors

```
protected SecurityException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

SecurityManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SecurityManager
   extends System.Object
```

Public Properties

```
public static boolean get_CheckExecutionRights();
public static void set_CheckExecutionRights(
   boolean value
);
public static boolean get_SecurityEnabled();
public static void set_SecurityEnabled(
   boolean value
);
```

Public Methods

```
public static boolean IsGranted(
   System.Security.IPermission perm
);
public static System.Security.Policy.PolicyLevel LoadPolicyLevelFromFile(
   System.String path,
   System.Security.PolicyLevelType type
);
public static System.Security.Policy.PolicyLevel LoadPolicyLevelFromString(
   System.String str,
   System.Security.PolicyLevelType type
);
public static System.Collections.IEnumerator PolicyHierarchy();
public static System.Security.PermissionSet ResolvePolicy(
   System.Security.Policy.Evidence evidence
);
public static System.Security.PermissionSet ResolvePolicy(
   System.Security.Policy.Evidence evidence,
   System.Security.PermissionSet reqdPset,
   System.Security.PermissionSet optPset,
   System.Security.PermissionSet denyPset,
   System.Security.PermissionSet& denied
);
public static System.Collections.IEnumerator ResolvePolicyGroups(
   System.Security.Policy.Evidence evidence
);
public static void SavePolicy();
public static void SavePolicyLevel(
   System.Security.Policy.PolicyLevel level
);
```

SecurityZone Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SecurityZone
```

Members

Internet

Intranet

MyComputer

NoZone

Trusted

Untrusted

SuppressUnmanagedCodeSecurityAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Method, Interface, AllowMultip
le = true, Inherited = false) */
public class SuppressUnmanagedCodeSecurityAttribute
   extends System.Attribute
```

Public Constructors

```
public SuppressUnmanagedCodeSecurityAttribute();
```

UnverifiableCodeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Module, AllowMultiple = true, Inherit
ed = false) */
public class UnverifiableCodeAttribute
   extends System.Attribute
```

Public Constructors

```
public UnverifiableCodeAttribute();
```

VerificationException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class VerificationException
   extends System.SystemException
```

Public Constructors

```
public VerificationException();
public VerificationException(
   System.String message
);
public VerificationException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected VerificationException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

XmlSyntaxException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class XmlSyntaxException
   extends System.SystemException
```

Public Constructors

```
public XmlSyntaxException();
public XmlSyntaxException(
   int lineNumber
);
public XmlSyntaxException(
   System.String message
);
public XmlSyntaxException(
   int lineNumber,
   System.String message
);
public XmlSyntaxException(
   System.String message,
   System.Exception inner
);
```

System.Security.Cryptography Namespace (Visual J# Syntax)

Classes

AsymmetricAlgorithm

AsymmetricKeyExchangeDeformatter

AsymmetricKeyExchangeFormatter

AsymmetricSignatureDeformatter

AsymmetricSignatureFormatter

CryptoAPITransform

CryptoConfig

CryptographicException

CryptographicUnexpectedOperationException

CryptoStream

CspParameters

DeriveBytes

DES

DESCryptoServiceProvider

DSA

DSACryptoServiceProvider

DSASignatureDeformatter

DSASignatureFormatter

FromBase64Transform

HashAlgorithm

HMACSHA1

KeyedHashAlgorithm

KeySizes

MACTripleDES

MaskGenerationMethod

MD5

MD5CryptoServiceProvider

PasswordDeriveBytes

PKCS1MaskGenerationMethod

RandomNumberGenerator

RC2

RC2CryptoServiceProvider

Rijndael

RijndaelManaged

RNGCryptoServiceProvider

RSA

RSACryptoServiceProvider

RSAOAEPKeyExchangeDeformatter

RSAOAEPKeyExchangeFormatter

RSAPKCS1KeyExchangeDeformatter

RSAPKCS1KeyExchangeFormatter

RSAPKCS1SignatureDeformatter

RSAPKCS1SignatureFormatter

SHA1

SHA1CryptoServiceProvider

SHA1Managed

SHA256

SHA256Managed

SHA384

SHA384Managed

SHA512

SHA512Managed

SignatureDescription

SymmetricAlgorithm

ToBase64Transform

TripleDES

TripleDESCryptoServiceProvider

Interfaces

ICryptoTransform

Structures

DSAParameters

RSAParameters

Enumerations

CipherMode

CryptoStreamMode

CspProviderFlags

FromBase64TransformMode

PaddingMode

AsymmetricAlgorithm Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class AsymmetricAlgorithm
   extends System.Object
   implements System.IDisposable
```

Public Properties

```
public abstract System.String get_KeyExchangeAlgorithm();
public int get_KeySize();
public void set_KeySize(
   int value
);
public System.Security.Cryptography.KeySizes[] get_LegalKeySizes();
public abstract System.String get_SignatureAlgorithm();
```

Public Methods

```
public void Clear();
public static System.Security.Cryptography.AsymmetricAlgorithm Create();
public static System.Security.Cryptography.AsymmetricAlgorithm Create(
   System.String algName
);
public abstract void FromXmlString(
   System.String xmlString
);
public abstract System.String ToXmlString(
   boolean includePrivateParameters
);
```

Protected Constructors

```
protected AsymmetricAlgorithm();
```

Protected Methods

```
protected abstract void Dispose(
   boolean disposing
);
```

Protected Fields

```
protected int KeySizeValue;
protected System.Security.Cryptography.KeySizes[] LegalKeySizesValue;
```

AsymmetricKeyExchangeDeformatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class AsymmetricKeyExchangeDeformatter
   extends System.Object
```

Public Constructors

```
public AsymmetricKeyExchangeDeformatter();
```

Public Properties

```
public abstract System.String get_Parameters();
public abstract void set_Parameters(
   System.String value
);
```

Public Methods

```
public abstract ubyte[] DecryptKeyExchange(
   ubyte[] rgb
);
public abstract void SetKey(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

AsymmetricKeyExchangeFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class AsymmetricKeyExchangeFormatter
   extends System.Object
```

Public Constructors

```
public AsymmetricKeyExchangeFormatter();
```

Public Properties

```
public abstract System.String get_Parameters();
```

Public Methods

```
public abstract ubyte[] CreateKeyExchange(
   ubyte[] data
);
public abstract ubyte[] CreateKeyExchange(
   ubyte[] data,
   System.Type symAlgType
);
public abstract void SetKey(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

AsymmetricSignatureDeformatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class AsymmetricSignatureDeformatter
   extends System.Object
```

Public Constructors

```
public AsymmetricSignatureDeformatter();
```

Public Methods

```
public abstract void SetHashAlgorithm(
   System.String strName
);
public abstract void SetKey(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
public abstract boolean VerifySignature(
   ubyte[] rgbHash,
   ubyte[] rgbSignature
);
public boolean VerifySignature(
   System.Security.Cryptography.HashAlgorithm hash,
   ubyte[] rgbSignature
);
```

AsymmetricSignatureFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class AsymmetricSignatureFormatter
   extends System.Object
```

Public Constructors

```
public AsymmetricSignatureFormatter();
```

Public Methods

```
public abstract ubyte[] CreateSignature(
   ubyte[] rgbHash
);
public ubyte[] CreateSignature(
   System.Security.Cryptography.HashAlgorithm hash
);
public abstract void SetHashAlgorithm(
   System.String strName
);
public abstract void SetKey(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

CipherMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CipherMode
```

Members

CBC

CFB

CTS

ECB

OFB

CryptoAPITransform Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CryptoAPITransform
   extends System.Object
   implements System.Security.Cryptography.ICryptoTransform, System.IDisposable
```

Public Properties

```
public final boolean get_CanReuseTransform();
public final boolean get_CanTransformMultipleBlocks();
public final int get_InputBlockSize();
public System.IntPtr get_KeyHandle();
public final int get_OutputBlockSize();
```

Public Methods

```
public void Clear();
public final int TransformBlock(
   ubyte[] inputBuffer,
   int inputOffset,
   int inputCount,
   ubyte[] outputBuffer,
   int outputOffset
);
public final ubyte[] TransformFinalBlock(
   ubyte[] inputBuffer,
   int inputOffset,
   int inputCount
);
```

Protected Methods

```
protected void Finalize();
```

CryptoConfig Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CryptoConfig
   extends System.Object
```

Public Constructors

```
public CryptoConfig();
```

Public Methods

```
public static System.Object CreateFromName(
   System.String name
);
public static System.Object CreateFromName(
   System.String name,
   System.Object[] args
);
public static ubyte[] EncodeOID(
   System.String str
);
public static System.String MapNameToOID(
   System.String name
);
```

CryptographicException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class CryptographicException
   extends System.SystemException
```

Public Constructors

```
public CryptographicException();
public CryptographicException(
   int hr
);
public CryptographicException(
   System.String message
);
public CryptographicException(
   System.String message,
   System.Exception inner
);
public CryptographicException(
   System.String format,
   System.String insert
);
```

Protected Constructors

```
protected CryptographicException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

CryptographicUnexpectedOperationException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class CryptographicUnexpectedOperationException
   extends System.Security.Cryptography.CryptographicException
```

Public Constructors

```
public CryptographicUnexpectedOperationException();
public CryptographicUnexpectedOperationException(
   System.String message
);
public CryptographicUnexpectedOperationException(
   System.String message,
   System.Exception inner
);
public CryptographicUnexpectedOperationException(
   System.String format,
   System.String insert
);
```

Protected Constructors

```
protected CryptographicUnexpectedOperationException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

CryptoStream Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CryptoStream
   extends System.IO.Stream
```

Public Constructors

```
public CryptoStream(
   System.IO.Stream stream,
   System.Security.Cryptography.ICryptoTransform transform,
   System.Security.Cryptography.CryptoStreamMode mode
);
```

Public Properties

```
public boolean get_CanRead();
public boolean get_CanSeek();
public boolean get_CanWrite();
public long get_Length();
public long get_Position();
public void set_Position(
   long value
);
```

Public Methods

```
public void Clear();
public void Close();
public void Flush();
public void FlushFinalBlock();
public int Read(
   ubyte[] buffer,
   int offset,
   int count
);
public long Seek(
   long offset,
   System.IO.SeekOrigin origin
);
public void SetLength(
   long value
);
public void Write(
   ubyte[] buffer,
   int offset,
   int count
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

```
protected void Finalize();
```

CryptoStreamMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CryptoStreamMode
```

Members

Read

Write

CspParameters Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CspParameters
   extends System.Object
```

Public Constructors

```
public CspParameters();
public CspParameters(
   int dwTypeIn
);
public CspParameters(
   int dwTypeIn,
   System.String strProviderNameIn
);
public CspParameters(
   int dwTypeIn,
   System.String strProviderNameIn,
   System.String strContainerNameIn
);
```

Public Properties

```
public System.Security.Cryptography.CspProviderFlags get_Flags();
public void set_Flags(
   System.Security.Cryptography.CspProviderFlags value
);
```

Public Fields

```
public System.String KeyContainerName;
public int KeyNumber;
public System.String ProviderName;
public int ProviderType;
```

CspProviderFlags Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum CspProviderFlags
```

Members

UseDefaultKeyContainer

UseMachineKeyStore

DeriveBytes Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class DeriveBytes
   extends System.Object
```

Public Methods

```
public abstract ubyte[] GetBytes(
   int cb
);
public abstract void Reset();
```

Protected Constructors

```
protected DeriveBytes();
```

DES Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class DES
   extends System.Security.Cryptography.SymmetricAlgorithm
```

Public Constructors

```
public DES();
```

Public Properties

```
public ubyte[] get_Key();
public void set_Key(
   ubyte[] value
);
```

Public Methods

```
public static System.Security.Cryptography.DES Create();
public static System.Security.Cryptography.DES Create(
   System.String algName
);
public static boolean IsSemiWeakKey(
   ubyte[] rgbKey
);
public static boolean IsWeakKey(
   ubyte[] rgbKey
);
```

DESCryptoServiceProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DESCryptoServiceProvider
   extends System.Security.Cryptography.DES
```

Public Constructors

```
public DESCryptoServiceProvider();
```

Public Methods

```
public System.Security.Cryptography.ICryptoTransform CreateDecryptor(
   ubyte[] rgbKey,
   ubyte[] rgbIV
);
public System.Security.Cryptography.ICryptoTransform CreateEncryptor(
   ubyte[] rgbKey,
   ubyte[] rgbIV
);
public void GenerateIV();
public void GenerateKey();
```

DSA Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class DSA
   extends System.Security.Cryptography.AsymmetricAlgorithm
```

Public Methods

```
public static System.Security.Cryptography.DSA Create();
public static System.Security.Cryptography.DSA Create(
   System.String algName
);
public abstract ubyte[] CreateSignature(
   ubyte[] rgbHash
);
public abstract System.Security.Cryptography.DSAParameters ExportParameters(
   boolean includePrivateParameters
);
public void FromXmlString(
   System.String xmlString
);
public abstract void ImportParameters(
   System.Security.Cryptography.DSAParameters parameters
);
public System.String ToXmlString(
   boolean includePrivateParameters
);
public abstract boolean VerifySignature(
   ubyte[] rgbHash,
   ubyte[] rgbSignature
);
```

DSACryptoServiceProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DSACryptoServiceProvider
   extends System.Security.Cryptography.DSA
```

Public Constructors

```
public DSACryptoServiceProvider();
public DSACryptoServiceProvider(
   int dwKeySize
);
public DSACryptoServiceProvider(
   System.Security.Cryptography.CspParameters parameters
);
public DSACryptoServiceProvider(
   int dwKeySize,
   System.Security.Cryptography.CspParameters parameters
);
```

Public Properties

```
public System.String get_KeyExchangeAlgorithm();
public int get_KeySize();
public System.Security.Cryptography.KeySizes[] get_LegalKeySizes();
public boolean get_PersistKeyInCsp();
public void set_PersistKeyInCsp(
   boolean value
);
public System.String get_SignatureAlgorithm();
```

Public Methods

```
public ubyte[] CreateSignature(
   ubyte[] rgbHash
);
public System.Security.Cryptography.DSAParameters ExportParameters(
   boolean includePrivateParameters
);
public void ImportParameters(
   System.Security.Cryptography.DSAParameters parameters
);
public ubyte[] SignData(
   ubyte[] buffer
);
public ubyte[] SignData(
   System.IO.Stream inputStream
);
public ubyte[] SignData(
   ubyte[] buffer,
   int offset,
   int count
);
public ubyte[] SignHash(
   ubyte[] rgbHash,
   System.String str
```

```
);
public boolean VerifyData(
   ubyte[] rgbData,
   ubyte[] rgbSignature
);
public boolean VerifyHash(
   ubyte[] rgbHash,
   System.String str,
   ubyte[] rgbSignature
);
public boolean VerifySignature(
   ubyte[] rgbHash,
   ubyte[] rgbSignature
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

DSAParameters Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class DSAParameters
   extends System.ValueType
```

Public Fields

```
public int Counter;
public ubyte[] G;
public ubyte[] J;
public ubyte[] P;
public ubyte[] Q;
public ubyte[] Seed;
/** @attribute System.NonSerializedAttribute() */
public ubyte[] X;
public ubyte[] Y;
```

DSASignatureDeformatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DSASignatureDeformatter
   extends System.Security.Cryptography.AsymmetricSignatureDeformatter
```

Public Constructors

```
public DSASignatureDeformatter();
public DSASignatureDeformatter(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

Public Methods

```
public void SetHashAlgorithm(
   System.String strName
);
public void SetKey(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
public boolean VerifySignature(
   ubyte[] rgbHash,
   ubyte[] rgbSignature
);
```

DSASignatureFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DSASignatureFormatter
   extends System.Security.Cryptography.AsymmetricSignatureFormatter
```

Public Constructors

```
public DSASignatureFormatter();
public DSASignatureFormatter(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

Public Methods

```
public ubyte[] CreateSignature(
   ubyte[] rgbHash
);
public void SetHashAlgorithm(
   System.String strName
);
public void SetKey(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

FromBase64Transform Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FromBase64Transform
   extends System.Object
   implements System.Security.Cryptography.ICryptoTransform, System.IDisposable
```

Public Constructors

```
public FromBase64Transform();
public FromBase64Transform(
   System.Security.Cryptography.FromBase64TransformMode whitespaces
);
```

Public Properties

```
public boolean get_CanReuseTransform();
public final boolean get_CanTransformMultipleBlocks();
public final int get_InputBlockSize();
public final int get_OutputBlockSize();
```

Public Methods

```
public void Clear();
public final int TransformBlock(
   ubyte[] inputBuffer,
   int inputOffset,
   int inputCount,
   ubyte[] outputBuffer,
   int outputOffset
);
public final ubyte[] TransformFinalBlock(
   ubyte[] inputBuffer,
   int inputOffset,
   int inputCount
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

FromBase64TransformMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum FromBase64TransformMode
```

Members

DoNotIgnoreWhiteSpaces

IgnoreWhiteSpaces

HashAlgorithm Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class HashAlgorithm
   extends System.Object
   implements System.Security.Cryptography.ICryptoTransform, System.IDisposable
```

Public Properties

```
public boolean get_CanReuseTransform();
public boolean get_CanTransformMultipleBlocks();
public ubyte[] get_Hash();
public int get_HashSize();
public int get_InputBlockSize();
public int get_OutputBlockSize();
```

Public Methods

```
public void Clear();
public ubyte[] ComputeHash(
   ubyte[] buffer
);
public ubyte[] ComputeHash(
   System.IO.Stream inputStream
);
public ubyte[] ComputeHash(
   ubyte[] buffer,
   int offset,
   int count
);
public static System.Security.Cryptography.HashAlgorithm Create();
public static System.Security.Cryptography.HashAlgorithm Create(
   System.String hashName
);
public abstract void Initialize();
public final int TransformBlock(
   ubyte[] inputBuffer,
   int inputOffset,
   int inputCount,
   ubyte[] outputBuffer,
   int outputOffset
);
public final ubyte[] TransformFinalBlock(
   ubyte[] inputBuffer,
   int inputOffset,
   int inputCount
);
```

Protected Constructors

```
protected HashAlgorithm();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected abstract void HashCore(
   ubyte[] array,
   int ibStart,
   int cbSize
);
protected abstract ubyte[] HashFinal();
```

Protected Fields

```
protected int HashSizeValue;
protected ubyte[] HashValue;
protected int State;
```

HMACSHA1 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HMACSHA1
   extends System.Security.Cryptography.KeyedHashAlgorithm
```

Public Constructors

```
public HMACSHA1();
public HMACSHA1(
   ubyte[] rgbKey
);
```

Public Properties

```
public System.String get_HashName();
public void set_HashName(
   System.String value
);
public ubyte[] get_Key();
public void set_Key(
   ubyte[] value
);
```

Public Methods

```
public void Initialize();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
protected void HashCore(
   ubyte[] rgb,
   int ib,
   int cb
);
protected ubyte[] HashFinal();
```

ICryptoTransform Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICryptoTransform
   implements System.IDisposable
```

Public Properties

```
boolean get_CanReuseTransform();
boolean get_CanTransformMultipleBlocks();
int get_InputBlockSize();
int get_OutputBlockSize();
```

Public Methods

```
int TransformBlock(
   ubyte[] inputBuffer,
   int inputOffset,
   int inputCount,
   ubyte[] outputBuffer,
   int outputOffset
);
ubyte[] TransformFinalBlock(
   ubyte[] inputBuffer,
   int inputOffset,
   int inputCount
);
```

KeyedHashAlgorithm Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class KeyedHashAlgorithm
   extends System.Security.Cryptography.HashAlgorithm
```

Public Properties

```
public ubyte[] get_Key();
public void set_Key(
   ubyte[] value
);
```

Public Methods

```
public static System.Security.Cryptography.KeyedHashAlgorithm Create();
public static System.Security.Cryptography.KeyedHashAlgorithm Create(
   System.String algName
);
```

Protected Constructors

```
protected KeyedHashAlgorithm();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

Protected Fields

```
protected ubyte[] KeyValue;
```

KeySizes Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class KeySizes
   extends System.Object
```

Public Constructors

```
public KeySizes(
   int minSize,
   int maxSize,
   int skipSize
);
```

Public Properties

```
public int get_MaxSize();
public int get_MinSize();
public int get_SkipSize();
```

MACTripleDES Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MACTripleDES
   extends System.Security.Cryptography.KeyedHashAlgorithm
```

Public Constructors

```
public MACTripleDES();
public MACTripleDES(
   ubyte[] rgbKey
);
public MACTripleDES(
   System.String strTripleDES,
   ubyte[] rgbKey
);
```

Public Methods

```
public void Initialize();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
protected void HashCore(
   ubyte[] rgbData,
   int ibStart,
   int cbSize
);
protected ubyte[] HashFinal();
```

MaskGenerationMethod Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class MaskGenerationMethod
   extends System.Object
```

Public Methods

```
public abstract ubyte[] GenerateMask(
   ubyte[] rgbSeed,
   int cbReturn
);
```

Protected Constructors

```
protected MaskGenerationMethod();
```

MD5 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class MD5
   extends System.Security.Cryptography.HashAlgorithm
```

Public Methods

```
public static System.Security.Cryptography.MD5 Create();
public static System.Security.Cryptography.MD5 Create(
   System.String algName
);
```

Protected Constructors

```
protected MD5();
```

MD5CryptoServiceProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MD5CryptoServiceProvider
   extends System.Security.Cryptography.MD5
```

Public Constructors

```
public MD5CryptoServiceProvider();
```

Public Methods

```
public void Initialize();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
protected void HashCore(
   ubyte[] rgb,
   int ibStart,
   int cbSize
);
protected ubyte[] HashFinal();
```

PaddingMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PaddingMode
```

Members

None

PKCS7

Zeros

PasswordDeriveBytes Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PasswordDeriveBytes
   extends System.Security.Cryptography.DeriveBytes
```

Public Constructors

```
public PasswordDeriveBytes(
   System.String strPassword,
   ubyte[] rgbSalt
);
public PasswordDeriveBytes(
   System.String strPassword,
   ubyte[] rgbSalt,
   System.Security.Cryptography.CspParameters cspParams
);
public PasswordDeriveBytes(
   System.String strPassword,
   ubyte[] rgbSalt,
   System.String strHashName,
   int iterations
);
public PasswordDeriveBytes(
   System.String strPassword,
   ubyte[] rgbSalt,
   System.String strHashName,
   int iterations,
   System.Security.Cryptography.CspParameters cspParams
);
```

Public Properties

```
public System.String get_HashName();
public void set_HashName(
   System.String value
);
public int get_IterationCount();
public void set_IterationCount(
   int value
);
public ubyte[] get_Salt();
public void set_Salt(
   ubyte[] value
);
```

Public Methods

```
public ubyte[] CryptDeriveKey(
   System.String algname,
   System.String alghashname,
   int keySize,
   ubyte[] rgbIV
);
public ubyte[] GetBytes(
   int cb
```

```
);
public void Reset();
```

Protected Methods

```
protected void Finalize();
```

PKCS1MaskGenerationMethod Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PKCS1MaskGenerationMethod
   extends System.Security.Cryptography.MaskGenerationMethod
```

Public Constructors

```
public PKCS1MaskGenerationMethod();
```

Public Properties

```
public System.String get_HashName();
public void set_HashName(
   System.String value
);
```

Public Methods

```
public ubyte[] GenerateMask(
   ubyte[] rgbSeed,
   int cbReturn
);
```

RandomNumberGenerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class RandomNumberGenerator
   extends System.Object
```

Public Constructors

```
public RandomNumberGenerator();
```

Public Methods

```
public static System.Security.Cryptography.RandomNumberGenerator Create();
public static System.Security.Cryptography.RandomNumberGenerator Create(
   System.String rngName
);
public abstract void GetBytes(
   ubyte[] data
);
public abstract void GetNonZeroBytes(
   ubyte[] data
);
```

RC2 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class RC2
   extends System.Security.Cryptography.SymmetricAlgorithm
```

Public Constructors

```
public RC2();
```

Public Properties

```
public int get_EffectiveKeySize();
public void set_EffectiveKeySize(
   int value
);
public int get_KeySize();
public void set_KeySize(
   int value
);
```

Public Methods

```
public static System.Security.Cryptography.RC2 Create();
public static System.Security.Cryptography.RC2 Create(
   System.String AlgName
);
```

Protected Fields

```
protected int EffectiveKeySizeValue;
```

RC2CryptoServiceProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RC2CryptoServiceProvider
   extends System.Security.Cryptography.RC2
```

Public Constructors

```
public RC2CryptoServiceProvider();
```

Public Properties

```
public int get_EffectiveKeySize();
public void set_EffectiveKeySize(
   int value
);
```

Public Methods

```
public System.Security.Cryptography.ICryptoTransform CreateDecryptor(
   ubyte[] rgbKey,
   ubyte[] rgbIV
);
public System.Security.Cryptography.ICryptoTransform CreateEncryptor(
   ubyte[] rgbKey,
   ubyte[] rgbIV
);
public void GenerateIV();
public void GenerateKey();
```

Rijndael Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class Rijndael
   extends System.Security.Cryptography.SymmetricAlgorithm
```

Public Constructors

```
public Rijndael();
```

Public Methods

```
public static System.Security.Cryptography.Rijndael Create();
public static System.Security.Cryptography.Rijndael Create(
   System.String algName
);
```

RijndaelManaged Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RijndaelManaged
   extends System.Security.Cryptography.Rijndael
```

Public Constructors

```
public RijndaelManaged();
```

Public Methods

```
public System.Security.Cryptography.ICryptoTransform CreateDecryptor(
   ubyte[] rgbKey,
   ubyte[] rgbIV
);
public System.Security.Cryptography.ICryptoTransform CreateEncryptor(
   ubyte[] rgbKey,
   ubyte[] rgbIV
);
public void GenerateIV();
public void GenerateKey();
```

RNGCryptoServiceProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RNGCryptoServiceProvider
   extends System.Security.Cryptography.RandomNumberGenerator
```

Public Constructors

```
public RNGCryptoServiceProvider();
public RNGCryptoServiceProvider(
   ubyte[] rgb
);
public RNGCryptoServiceProvider(
   System.Security.Cryptography.CspParameters cspParams
);
public RNGCryptoServiceProvider(
   System.String str
);
```

Public Methods

```
public void GetBytes(
   ubyte[] data
);
public void GetNonZeroBytes(
   ubyte[] data
);
```

Protected Methods

```
protected void Finalize();
```

RSA Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class RSA
   extends System.Security.Cryptography.AsymmetricAlgorithm
```

Public Constructors

```
public RSA();
```

Public Methods

```
public static System.Security.Cryptography.RSA Create();
public static System.Security.Cryptography.RSA Create(
   System.String algName
);
public abstract ubyte[] DecryptValue(
   ubyte[] rgb
);
public abstract ubyte[] EncryptValue(
   ubyte[] rgb
);
public abstract System.Security.Cryptography.RSAParameters ExportParameters(
   boolean includePrivateParameters
);
public void FromXmlString(
   System.String xmlString
);
public abstract void ImportParameters(
   System.Security.Cryptography.RSAParameters parameters
);
public System.String ToXmlString(
   boolean includePrivateParameters
);
```

RSACryptoServiceProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RSACryptoServiceProvider
   extends System.Security.Cryptography.RSA
```

Public Constructors

```
public RSACryptoServiceProvider();
public RSACryptoServiceProvider(
   int dwKeySize
);
public RSACryptoServiceProvider(
   System.Security.Cryptography.CspParameters parameters
);
public RSACryptoServiceProvider(
   int dwKeySize,
   System.Security.Cryptography.CspParameters parameters
);
```

Public Properties

```
public System.String get_KeyExchangeAlgorithm();
public int get_KeySize();
public boolean get_PersistKeyInCsp();
public void set_PersistKeyInCsp(
   boolean value
);
public System.String get_SignatureAlgorithm();
```

Public Methods

```
public ubyte[] Decrypt(
   ubyte[] rgb,
   boolean fOAEP
);
public ubyte[] DecryptValue(
   ubyte[] rgb
);
public ubyte[] Encrypt(
   ubyte[] rgb,
   boolean fOAEP
);
public ubyte[] EncryptValue(
   ubyte[] rgb
);
public System.Security.Cryptography.RSAParameters ExportParameters(
   boolean includePrivateParameters
);
public void ImportParameters(
   System.Security.Cryptography.RSAParameters parameters
);
public ubyte[] SignData(
   ubyte[] buffer,
   System.Object halg
);
```

```
public ubyte[] SignData(
   System.IO.Stream inputStream,
   System.Object halg
);
public ubyte[] SignData(
   ubyte[] buffer,
   int offset,
   int count,
   System.Object halg
);
public ubyte[] SignHash(
   ubyte[] rgbHash,
   System.String str
);
public boolean VerifyData(
   ubyte[] buffer,
   System.Object halg,
   ubyte[] signature
);
public boolean VerifyHash(
   ubyte[] rgbHash,
   System.String str,
   ubyte[] rgbSignature
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

RSAOAEPKeyExchangeDeformatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RSAOAEPKeyExchangeDeformatter
   extends System.Security.Cryptography.AsymmetricKeyExchangeDeformatter
```

Public Constructors

```
public RSAOAEPKeyExchangeDeformatter();
public RSAOAEPKeyExchangeDeformatter(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

Public Properties

```
public System.String get_Parameters();
public void set_Parameters(
   System.String value
);
```

Public Methods

```
public ubyte[] DecryptKeyExchange(
   ubyte[] rgbData
);
public void SetKey(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

RSAOAEPKeyExchangeFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RSAOAEPKeyExchangeFormatter
   extends System.Security.Cryptography.AsymmetricKeyExchangeFormatter
```

Public Constructors

```
public RSAOAEPKeyExchangeFormatter();
public RSAOAEPKeyExchangeFormatter(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

Public Properties

```
public ubyte[] get_Parameter();
public void set_Parameter(
   ubyte[] value
);
public System.String get_Parameters();
public System.Security.Cryptography.RandomNumberGenerator get_Rng();
public void set_Rng(
   System.Security.Cryptography.RandomNumberGenerator value
);
```

Public Methods

```
public ubyte[] CreateKeyExchange(
   ubyte[] rgbData
);
public ubyte[] CreateKeyExchange(
   ubyte[] rgbData,
   System.Type symAlgType
);
public void SetKey(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

RSAParameters Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RSAParameters
   extends System.ValueType
```

Public Fields

```
/** @attribute System.NonSerializedAttribute() */
public ubyte[] D;
/** @attribute System.NonSerializedAttribute() */
public ubyte[] DP;
/** @attribute System.NonSerializedAttribute() */
public ubyte[] DQ;
public ubyte[] Exponent;
/** @attribute System.NonSerializedAttribute() */
public ubyte[] InverseQ;
public ubyte[] Modulus;
/** @attribute System.NonSerializedAttribute() */
public ubyte[] P;
/** @attribute System.NonSerializedAttribute() */
public ubyte[] Q;
```

RSAPKCS1KeyExchangeDeformatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RSAPKCS1KeyExchangeDeformatter
   extends System.Security.Cryptography.AsymmetricKeyExchangeDeformatter
```

Public Constructors

```
public RSAPKCS1KeyExchangeDeformatter();
public RSAPKCS1KeyExchangeDeformatter(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

Public Properties

```
public System.String get_Parameters();
public void set_Parameters(
   System.String value
);
public System.Security.Cryptography.RandomNumberGenerator get_RNG();
public void set_RNG(
   System.Security.Cryptography.RandomNumberGenerator value
);
```

Public Methods

```
public ubyte[] DecryptKeyExchange(
   ubyte[] rgbIn
);
public void SetKey(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

RSAPKCS1KeyExchangeFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RSAPKCS1KeyExchangeFormatter
   extends System.Security.Cryptography.AsymmetricKeyExchangeFormatter
```

Public Constructors

```
public RSAPKCS1KeyExchangeFormatter();
public RSAPKCS1KeyExchangeFormatter(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

Public Properties

```
public System.String get_Parameters();
public System.Security.Cryptography.RandomNumberGenerator get_Rng();
public void set_Rng(
   System.Security.Cryptography.RandomNumberGenerator value
);
```

Public Methods

```
public ubyte[] CreateKeyExchange(
   ubyte[] rgbData
);
public ubyte[] CreateKeyExchange(
   ubyte[] rgbData,
   System.Type symAlgType
);
public void SetKey(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

RSAPKCS1SignatureDeformatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RSAPKCS1SignatureDeformatter
   extends System.Security.Cryptography.AsymmetricSignatureDeformatter
```

Public Constructors

```
public RSAPKCS1SignatureDeformatter();
public RSAPKCS1SignatureDeformatter(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

Public Methods

```
public void SetHashAlgorithm(
   System.String strName
);
public void SetKey(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
public boolean VerifySignature(
   ubyte[] rgbHash,
   ubyte[] rgbSignature
);
```

RSAPKCS1SignatureFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RSAPKCS1SignatureFormatter
   extends System.Security.Cryptography.AsymmetricSignatureFormatter
```

Public Constructors

```
public RSAPKCS1SignatureFormatter();
public RSAPKCS1SignatureFormatter(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

Public Methods

```
public ubyte[] CreateSignature(
   ubyte[] rgbHash
);
public void SetHashAlgorithm(
   System.String strName
);
public void SetKey(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

SHA1 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class SHA1
   extends System.Security.Cryptography.HashAlgorithm
```

Public Methods

```
public static System.Security.Cryptography.SHA1 Create();
public static System.Security.Cryptography.SHA1 Create(
   System.String hashName
);
```

Protected Constructors

```
protected SHA1();
```

SHA1CryptoServiceProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SHA1CryptoServiceProvider
   extends System.Security.Cryptography.SHA1
```

Public Constructors

```
public SHA1CryptoServiceProvider();
```

Public Methods

```
public void Initialize();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
protected void HashCore(
   ubyte[] rgb,
   int ibStart,
   int cbSize
);
protected ubyte[] HashFinal();
```

SHA1Managed Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SHA1Managed
   extends System.Security.Cryptography.SHA1
```

Public Constructors

```
public SHA1Managed();
```

Public Methods

```
public void Initialize();
```

Protected Methods

```
protected void HashCore(
   ubyte[] rgb,
   int ibStart,
   int cbSize
);
protected ubyte[] HashFinal();
```

SHA256 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class SHA256
   extends System.Security.Cryptography.HashAlgorithm
```

Public Constructors

```
public SHA256();
```

Public Methods

```
public static System.Security.Cryptography.SHA256 Create();
public static System.Security.Cryptography.SHA256 Create(
   System.String hashName
);
```

SHA256Managed Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SHA256Managed
   extends System.Security.Cryptography.SHA256
```

Public Constructors

```
public SHA256Managed();
```

Public Methods

```
public void Initialize();
```

Protected Methods

```
protected void HashCore(
   ubyte[] rgb,
   int ibStart,
   int cbSize
);
protected ubyte[] HashFinal();
```

SHA384 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class SHA384
   extends System.Security.Cryptography.HashAlgorithm
```

Public Constructors

```
public SHA384();
```

Public Methods

```
public static System.Security.Cryptography.SHA384 Create();
public static System.Security.Cryptography.SHA384 Create(
   System.String hashName
);
```

SHA384Managed Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SHA384Managed
   extends System.Security.Cryptography.SHA384
```

Public Constructors

```
public SHA384Managed();
```

Public Methods

```
public void Initialize();
```

Protected Methods

```
protected void HashCore(
   ubyte[] rgb,
   int ibStart,
   int cbSize
);
protected ubyte[] HashFinal();
```

SHA512 Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class SHA512
   extends System.Security.Cryptography.HashAlgorithm
```

Public Constructors

```
public SHA512();
```

Public Methods

```
public static System.Security.Cryptography.SHA512 Create();
public static System.Security.Cryptography.SHA512 Create(
   System.String hashName
);
```

SHA512Managed Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SHA512Managed
   extends System.Security.Cryptography.SHA512
```

Public Constructors

```
public SHA512Managed();
```

Public Methods

```
public void Initialize();
```

Protected Methods

```
protected void HashCore(
   ubyte[] rgb,
   int ibStart,
   int cbSize
);
protected ubyte[] HashFinal();
```

SignatureDescription Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SignatureDescription
   extends System.Object
```

Public Constructors

```
public SignatureDescription();
public SignatureDescription(
   System.Security.SecurityElement el
);
```

Public Properties

```
public System.String get_DeformatterAlgorithm();
public void set_DeformatterAlgorithm(
   System.String value
);
public System.String get_DigestAlgorithm();
public void set_DigestAlgorithm(
   System.String value
);
public System.String get_FormatterAlgorithm();
public void set_FormatterAlgorithm(
   System.String value
);
public System.String get_KeyAlgorithm();
public void set_KeyAlgorithm(
   System.String value
);
```

Public Methods

```
public System.Security.Cryptography.AsymmetricSignatureDeformatter CreateDeformatter(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
public System.Security.Cryptography.HashAlgorithm CreateDigest();
public System.Security.Cryptography.AsymmetricSignatureFormatter CreateFormatter(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
```

SymmetricAlgorithm Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class SymmetricAlgorithm
   extends System.Object
   implements System.IDisposable
```

Public Constructors

```
public SymmetricAlgorithm();
```

Public Properties

```
public int get_BlockSize();
public void set_BlockSize(
   int value
);
public int get_FeedbackSize();
public void set_FeedbackSize(
   int value
);
public ubyte[] get_IV();
public void set_IV(
   ubyte[] value
);
public ubyte[] get_Key();
public void set_Key(
   ubyte[] value
);
public int get_KeySize();
public void set_KeySize(
   int value
);
public System.Security.Cryptography.KeySizes[] get_LegalBlockSizes();
public System.Security.Cryptography.KeySizes[] get_LegalKeySizes();
public System.Security.Cryptography.CipherMode get_Mode();
public void set_Mode(
   System.Security.Cryptography.CipherMode value
);
public System.Security.Cryptography.PaddingMode get_Padding();
public void set_Padding(
   System.Security.Cryptography.PaddingMode value
);
```

Public Methods

```
public void Clear();
public static System.Security.Cryptography.SymmetricAlgorithm Create();
public static System.Security.Cryptography.SymmetricAlgorithm Create(
   System.String algName
);
public System.Security.Cryptography.ICryptoTransform CreateDecryptor();
public abstract System.Security.Cryptography.ICryptoTransform CreateDecryptor(
   ubyte[] rgbKey,
   ubyte[] rgbIV
);
```

```
public System.Security.Cryptography.ICryptoTransform CreateEncryptor();
public abstract System.Security.Cryptography.ICryptoTransform CreateEncryptor(
   ubyte[] rgbKey,
   ubyte[] rgbIV
);
public abstract void GenerateIV();
public abstract void GenerateKey();
public boolean ValidKeySize(
   int bitLength
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

Protected Fields

```
protected int BlockSizeValue;
protected int FeedbackSizeValue;
protected ubyte[] IVValue;
protected int KeySizeValue;
protected ubyte[] KeyValue;
protected System.Security.Cryptography.KeySizes[] LegalBlockSizesValue;
protected System.Security.Cryptography.KeySizes[] LegalKeySizesValue;
protected System.Security.Cryptography.CipherMode ModeValue;
protected System.Security.Cryptography.PaddingMode PaddingValue;
```

ToBase64Transform Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ToBase64Transform
   extends System.Object
   implements System.Security.Cryptography.ICryptoTransform, System.IDisposable
```

Public Constructors

```
public ToBase64Transform();
```

Public Properties

```
public boolean get_CanReuseTransform();
public final boolean get_CanTransformMultipleBlocks();
public final int get_InputBlockSize();
public final int get_OutputBlockSize();
```

Public Methods

```
public void Clear();
public final int TransformBlock(
   ubyte[] inputBuffer,
   int inputOffset,
   int inputCount,
   ubyte[] outputBuffer,
   int outputOffset
);
public final ubyte[] TransformFinalBlock(
   ubyte[] inputBuffer,
   int inputOffset,
   int inputCount
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

TripleDES Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class TripleDES
   extends System.Security.Cryptography.SymmetricAlgorithm
```

Public Constructors

```
public TripleDES();
```

Public Properties

```
public ubyte[] get_Key();
public void set_Key(
   ubyte[] value
);
```

Public Methods

```
public static System.Security.Cryptography.TripleDES Create();
public static System.Security.Cryptography.TripleDES Create(
   System.String str
);
public static boolean IsWeakKey(
   ubyte[] rgbKey
);
```

TripleDESCryptoServiceProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TripleDESCryptoServiceProvider
   extends System.Security.Cryptography.TripleDES
```

Public Constructors

```
public TripleDESCryptoServiceProvider();
```

Public Methods

```
public System.Security.Cryptography.ICryptoTransform CreateDecryptor(
   ubyte[] rgbKey,
   ubyte[] rgbIV
);
public System.Security.Cryptography.ICryptoTransform CreateEncryptor(
   ubyte[] rgbKey,
   ubyte[] rgbIV
);
public void GenerateIV();
public void GenerateKey();
```

System.Security.Cryptography.X509Certificates Namespace (Visual J# Syntax)

Classes

X509Certificate

X509CertificateCollection

X509CertificateCollection.X509CertificateEnumerator

X509Certificate Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class X509Certificate
   extends System.Object
```

Public Constructors

```
public X509Certificate(
   ubyte[] data
);
public X509Certificate(
   System.IntPtr handle
);
public X509Certificate(
   System.Security.Cryptography.X509Certificates.X509Certificate cert
);
```

Public Methods

```
public static System.Security.Cryptography.X509Certificates.X509Certificate CreateFromCertFil
e(
   System.String filename
);
public static System.Security.Cryptography.X509Certificates.X509Certificate CreateFromSignedF
ile(
   System.String filename
);
public boolean Equals(
   System.Security.Cryptography.X509Certificates.X509Certificate other
);
public ubyte[] GetCertHash();
public System.String GetCertHashString();
public System.String GetEffectiveDateString();
public System.String GetExpirationDateString();
public System.String GetFormat();
public int GetHashCode();
public System.String GetIssuerName();
public System.String GetKeyAlgorithm();
public ubyte[] GetKeyAlgorithmParameters();
public System.String GetKeyAlgorithmParametersString();
public System.String GetName();
public ubyte[] GetPublicKey();
public System.String GetPublicKeyString();
public ubyte[] GetRawCertData();
public System.String GetRawCertDataString();
public ubyte[] GetSerialNumber();
public System.String GetSerialNumberString();
public System.String ToString();
public System.String ToString(
   boolean fVerbose
);
```

X509CertificateCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class X509CertificateCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public X509CertificateCollection();
public X509CertificateCollection(
   System.Security.Cryptography.X509Certificates.X509CertificateCollection value
);
public X509CertificateCollection(
   System.Security.Cryptography.X509Certificates.X509Certificate[] value
);
```

Public Properties

```
public System.Security.Cryptography.X509Certificates.X509Certificate get_Item(
   int index
);
public void set_Item(
   int index,
   System.Security.Cryptography.X509Certificates.X509Certificate value
);
```

Public Methods

```
public int Add(
   System.Security.Cryptography.X509Certificates.X509Certificate value
);
public void AddRange(
   System.Security.Cryptography.X509Certificates.X509CertificateCollection value
);
public void AddRange(
   System.Security.Cryptography.X509Certificates.X509Certificate[] value
);
public boolean Contains(
   System.Security.Cryptography.X509Certificates.X509Certificate value
);
public void CopyTo(
   System.Security.Cryptography.X509Certificates.X509Certificate[] array,
   int index
);
public System.Security.Cryptography.X509Certificates.X509CertificateCollection+X509Certificat
eEnumerator GetEnumerator();
public int GetHashCode();
public int IndexOf(
   System.Security.Cryptography.X509Certificates.X509Certificate value
);
public void Insert(
   int index,
   System.Security.Cryptography.X509Certificates.X509Certificate value
);
```

```
public void Remove(
   System.Security.Cryptography.X509Certificates.X509Certificate value
);
```

X509CertificateCollection.X509CertificateEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class X509CertificateEnumerator
   extends System.Object
   implements System.Collections.IEnumerator
```

Public Constructors

```
public X509CertificateEnumerator(
   System.Security.Cryptography.X509Certificates.X509CertificateCollection mappings
);
```

Public Properties

```
public System.Security.Cryptography.X509Certificates.X509Certificate get_Current();
```

Public Methods

```
public boolean MoveNext();
public void Reset();
```

System.Security.Cryptography.Xml Namespace (Visual J# Syntax)

Classes

DataObject

DSAKeyValue

KeyInfo

KeyInfoClause

KeyInfoName

KeyInfoNode

KeyInfoRetrievalMethod

KeyInfoX509Data

Reference

RSAKeyValue

Signature

SignedInfo

SignedXml

Transform

TransformChain

XmlDsigBase64Transform

XmlDsigC14NTransform

XmlDsigC14NWithCommentsTransform

XmlDsigEnvelopedSignatureTransform

XmlDsigXPathTransform

XmlDsigXsltTransform

DataObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataObject
   extends System.Object
```

Public Constructors

```
public DataObject();
public DataObject(
   System.String id,
   System.String mimeType,
   System.String encoding,
   System.Xml.XmlElement data
);
```

Public Properties

```
public System.Xml.XmlNodeList get_Data();
public void set_Data(
   System.Xml.XmlNodeList value
);
public System.String get_Encoding();
public void set_Encoding(
   System.String value
);
public System.String get_Id();
public void set_Id(
   System.String value
);
public System.String get_MimeType();
public void set_MimeType(
   System.String value
);
```

Public Methods

```
public System.Xml.XmlElement GetXml();
public void LoadXml(
   System.Xml.XmlElement value
);
```

DSAKeyValue Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DSAKeyValue
   extends System.Security.Cryptography.Xml.KeyInfoClause
```

Public Constructors

```
public DSAKeyValue();
public DSAKeyValue(
   System.Security.Cryptography.DSA key
);
```

Public Properties

```
public System.Security.Cryptography.DSA get_Key();
public void set_Key(
   System.Security.Cryptography.DSA value
);
```

Public Methods

```
public System.Xml.XmlElement GetXml();
public void LoadXml(
   System.Xml.XmlElement value
);
```

KeyInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class KeyInfo
   extends System.Object
   implements System.Collections.IEnumerable
```

Public Constructors

```
public KeyInfo();
```

Public Properties

```
public int get_Count();
public System.String get_Id();
public void set_Id(
   System.String value
);
```

Public Methods

```
public void AddClause(
   System.Security.Cryptography.Xml.KeyInfoClause clause
);
public final System.Collections.IEnumerator GetEnumerator();
public System.Collections.IEnumerator GetEnumerator(
   System.Type requestedObjectType
);
public System.Xml.XmlElement GetXml();
public void LoadXml(
   System.Xml.XmlElement value
);
```

KeyInfoClause Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class KeyInfoClause
   extends System.Object
```

Public Constructors

```
public KeyInfoClause();
```

Public Methods

```
public abstract System.Xml.XmlElement GetXml();
public abstract void LoadXml(
   System.Xml.XmlElement element
);
```

KeyInfoName Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class KeyInfoName
   extends System.Security.Cryptography.Xml.KeyInfoClause
```

Public Constructors

```
public KeyInfoName();
```

Public Properties

```
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Methods

```
public System.Xml.XmlElement GetXml();
public void LoadXml(
   System.Xml.XmlElement value
);
```

KeyInfoNode Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class KeyInfoNode
   extends System.Security.Cryptography.Xml.KeyInfoClause
```

Public Constructors

```
public KeyInfoNode();
public KeyInfoNode(
   System.Xml.XmlElement node
);
```

Public Properties

```
public System.Xml.XmlElement get_Value();
public void set_Value(
   System.Xml.XmlElement value
);
```

Public Methods

```
public System.Xml.XmlElement GetXml();
public void LoadXml(
   System.Xml.XmlElement value
);
```

KeyInfoRetrievalMethod Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class KeyInfoRetrievalMethod
   extends System.Security.Cryptography.Xml.KeyInfoClause
```

Public Constructors

```
public KeyInfoRetrievalMethod();
public KeyInfoRetrievalMethod(
   System.String strUri
);
```

Public Properties

```
public System.String get_Uri();
public void set_Uri(
   System.String value
);
```

Public Methods

```
public System.Xml.XmlElement GetXml();
public void LoadXml(
   System.Xml.XmlElement value
);
```

KeyInfoX509Data Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class KeyInfoX509Data
   extends System.Security.Cryptography.Xml.KeyInfoClause
```

Public Constructors

```
public KeyInfoX509Data();
public KeyInfoX509Data(
   ubyte[] rgbCert
);
public KeyInfoX509Data(
   System.Security.Cryptography.X509Certificates.X509Certificate cert
);
```

Public Properties

```
public System.Collections.ArrayList get_Certificates();
public ubyte[] get_CRL();
public void set_CRL(
   ubyte[] value
);
public System.Collections.ArrayList get_IssuerSerials();
public System.Collections.ArrayList get_SubjectKeyIds();
public System.Collections.ArrayList get_SubjectNames();
```

Public Methods

```
public void AddCertificate(
   System.Security.Cryptography.X509Certificates.X509Certificate certificate
);
public void AddIssuerSerial(
   System.String issuerName,
   System.String serialNumber
);
public void AddSubjectKeyId(
   ubyte[] subjectKeyId
);
public void AddSubjectName(
   System.String subjectName
);
public System.Xml.XmlElement GetXml();
public void LoadXml(
   System.Xml.XmlElement element
);
```

Reference Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Reference
   extends System.Object
```

Public Constructors

```
public Reference();
public Reference(
   System.IO.Stream stream
);
public Reference(
   System.String uri
);
```

Public Properties

```
public System.String get_DigestMethod();
public void set_DigestMethod(
   System.String value
);
public ubyte[] get_DigestValue();
public void set_DigestValue(
   ubyte[] value
);
public System.String get_Id();
public void set_Id(
   System.String value
);
public System.Security.Cryptography.Xml.TransformChain get_TransformChain();
public System.String get_Type();
public void set_Type(
   System.String value
);
public System.String get_Uri();
public void set_Uri(
   System.String value
);
```

Public Methods

```
public void AddTransform(
   System.Security.Cryptography.Xml.Transform transform
);
public System.Xml.XmlElement GetXml();
public void LoadXml(
   System.Xml.XmlElement value
);
```

RSAKeyValue Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RSAKeyValue
   extends System.Security.Cryptography.Xml.KeyInfoClause
```

Public Constructors

```
public RSAKeyValue();
public RSAKeyValue(
   System.Security.Cryptography.RSA key
);
```

Public Properties

```
public System.Security.Cryptography.RSA get_Key();
public void set_Key(
   System.Security.Cryptography.RSA value
);
```

Public Methods

```
public System.Xml.XmlElement GetXml();
public void LoadXml(
   System.Xml.XmlElement value
);
```

Signature Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Signature
   extends System.Object
```

Public Constructors

```
public Signature();
```

Public Properties

```
public System.String get_Id();
public void set_Id(
   System.String value
);
public System.Security.Cryptography.Xml.KeyInfo get_KeyInfo();
public void set_KeyInfo(
   System.Security.Cryptography.Xml.KeyInfo value
);
public System.Collections.IList get_ObjectList();
public void set_ObjectList(
   System.Collections.IList value
);
public ubyte[] get_SignatureValue();
public void set_SignatureValue(
   ubyte[] value
);
public System.Security.Cryptography.Xml.SignedInfo get_SignedInfo();
public void set_SignedInfo(
   System.Security.Cryptography.Xml.SignedInfo value
);
```

Public Methods

```
public void AddObject(
   System.Security.Cryptography.Xml.DataObject dataObject
);
public System.Xml.XmlElement GetXml();
public void LoadXml(
   System.Xml.XmlElement value
);
```

SignedInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SignedInfo
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public SignedInfo();
```

Public Properties

```
public System.String get_CanonicalizationMethod();
public void set_CanonicalizationMethod(
   System.String value
);
public final int get_Count();
public System.String get_Id();
public void set_Id(
   System.String value
);
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Collections.ArrayList get_References();
public System.String get_SignatureLength();
public void set_SignatureLength(
   System.String value
);
public System.String get_SignatureMethod();
public void set_SignatureMethod(
   System.String value
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public void AddReference(
   System.Security.Cryptography.Xml.Reference reference
);
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public System.Xml.XmlElement GetXml();
public void LoadXml(
   System.Xml.XmlElement value
);
```

SignedXml Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SignedXml
   extends System.Object
```

Public Constructors

```
public SignedXml();
public SignedXml(
   System.Xml.XmlDocument document
);
public SignedXml(
   System.Xml.XmlElement elem
);
```

Public Properties

```
public System.Security.Cryptography.Xml.KeyInfo get_KeyInfo();
public void set_KeyInfo(
   System.Security.Cryptography.Xml.KeyInfo value
);
public System.Security.Cryptography.Xml.Signature get_Signature();
public System.String get_SignatureLength();
public System.String get_SignatureMethod();
public ubyte[] get_SignatureValue();
public System.Security.Cryptography.Xml.SignedInfo get_SignedInfo();
public System.Security.Cryptography.AsymmetricAlgorithm get_SigningKey();
public void set_SigningKey(
   System.Security.Cryptography.AsymmetricAlgorithm value
);
public System.String get_SigningKeyName();
public void set_SigningKeyName(
   System.String value
);
```

Public Methods

```
public void AddObject(
   System.Security.Cryptography.Xml.DataObject dataObject
);
public void AddReference(
   System.Security.Cryptography.Xml.Reference reference
);
public boolean CheckSignature();
public boolean CheckSignature(
   System.Security.Cryptography.AsymmetricAlgorithm key
);
public boolean CheckSignature(
   System.Security.Cryptography.KeyedHashAlgorithm macAlg
);
public boolean CheckSignatureReturningKey(
   System.Security.Cryptography.AsymmetricAlgorithm& signingKey
);
public void ComputeSignature();
public void ComputeSignature(
```

```
   System.Security.Cryptography.KeyedHashAlgorithm macAlg
);
public System.Xml.XmlElement GetIdElement(
   System.Xml.XmlDocument document,
   System.String idValue
);
public System.Xml.XmlElement GetXml();
public void LoadXml(
   System.Xml.XmlElement value
);
```

Public Fields

```
public static System.String XmlDsigCanonicalizationUrl;
public static System.String XmlDsigCanonicalizationWithCommentsUrl;
public static System.String XmlDsigDSAUrl;
public static System.String XmlDsigHMACSHA1Url;
public static System.String XmlDsigMinimalCanonicalizationUrl;
public static System.String XmlDsigNamespaceUrl;
public static System.String XmlDsigRSASHA1Url;
public static System.String XmlDsigSHA1Url;
```

Protected Methods

```
protected System.Security.Cryptography.AsymmetricAlgorithm GetPublicKey();
```

Protected Fields

```
protected System.Security.Cryptography.Xml.Signature m_signature;
protected System.String m_strSigningKeyName;
```

Transform Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class Transform
   extends System.Object
```

Public Constructors

```
public Transform();
```

Public Properties

```
public System.String get_Algorithm();
public void set_Algorithm(
   System.String value
);
public abstract System.Type[] get_InputTypes();
public abstract System.Type[] get_OutputTypes();
```

Public Methods

```
public abstract System.Object GetOutput();
public abstract System.Object GetOutput(
   System.Type type
);
public System.Xml.XmlElement GetXml();
public abstract void LoadInnerXml(
   System.Xml.XmlNodeList nodeList
);
public abstract void LoadInput(
   System.Object obj
);
```

Protected Methods

```
protected abstract System.Xml.XmlNodeList GetInnerXml();
```

TransformChain Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class TransformChain
   extends System.Object
```

Public Constructors

```
public TransformChain();
```

Public Properties

```
public int get_Count();
public System.Security.Cryptography.Xml.Transform get_Item(
   int index
);
```

Public Methods

```
public void Add(
   System.Security.Cryptography.Xml.Transform transform
);
public System.Collections.IEnumerator GetEnumerator();
```

XmlDsigBase64Transform Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlDsigBase64Transform
   extends System.Security.Cryptography.Xml.Transform
```

Public Constructors

```
public XmlDsigBase64Transform();
```

Public Properties

```
public System.Type[] get_InputTypes();
public System.Type[] get_OutputTypes();
```

Public Methods

```
public System.Object GetOutput();
public System.Object GetOutput(
   System.Type type
);
public void LoadInnerXml(
   System.Xml.XmlNodeList nodeList
);
public void LoadInput(
   System.Object obj
);
```

Protected Methods

```
protected System.Xml.XmlNodeList GetInnerXml();
```

XmlDsigC14NTransform Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlDsigC14NTransform
   extends System.Security.Cryptography.Xml.Transform
```

Public Constructors

```
public XmlDsigC14NTransform();
public XmlDsigC14NTransform(
   boolean includeComments
);
```

Public Properties

```
public System.Type[] get_InputTypes();
public System.Type[] get_OutputTypes();
```

Public Methods

```
public System.Object GetOutput();
public System.Object GetOutput(
   System.Type type
);
public void LoadInnerXml(
   System.Xml.XmlNodeList nodeList
);
public void LoadInput(
   System.Object obj
);
```

Protected Methods

```
protected System.Xml.XmlNodeList GetInnerXml();
```

XmlDsigC14NWithCommentsTransform Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlDsigC14NWithCommentsTransform
   extends System.Security.Cryptography.Xml.XmlDsigC14NTransform
```

Public Constructors

```
public XmlDsigC14NWithCommentsTransform();
```

XmlDsigEnvelopedSignatureTransform Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlDsigEnvelopedSignatureTransform
   extends System.Security.Cryptography.Xml.Transform
```

Public Constructors

```
public XmlDsigEnvelopedSignatureTransform();
public XmlDsigEnvelopedSignatureTransform(
   boolean includeComments
);
```

Public Properties

```
public System.Type[] get_InputTypes();
public System.Type[] get_OutputTypes();
```

Public Methods

```
public System.Object GetOutput();
public System.Object GetOutput(
   System.Type type
);
public void LoadInnerXml(
   System.Xml.XmlNodeList nodeList
);
public void LoadInput(
   System.Object obj
);
```

Protected Methods

```
protected System.Xml.XmlNodeList GetInnerXml();
```

XmlDsigXPathTransform Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlDsigXPathTransform
   extends System.Security.Cryptography.Xml.Transform
```

Public Constructors

```
public XmlDsigXPathTransform();
```

Public Properties

```
public System.Type[] get_InputTypes();
public System.Type[] get_OutputTypes();
```

Public Methods

```
public System.Object GetOutput();
public System.Object GetOutput(
   System.Type type
);
public void LoadInnerXml(
   System.Xml.XmlNodeList nodeList
);
public void LoadInput(
   System.Object obj
);
```

Protected Methods

```
protected System.Xml.XmlNodeList GetInnerXml();
```

XmlDsigXsltTransform Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlDsigXsltTransform
   extends System.Security.Cryptography.Xml.Transform
```

Public Constructors

```
public XmlDsigXsltTransform();
public XmlDsigXsltTransform(
   boolean includeComments
);
```

Public Properties

```
public System.Type[] get_InputTypes();
public System.Type[] get_OutputTypes();
```

Public Methods

```
public System.Object GetOutput();
public System.Object GetOutput(
   System.Type type
);
public void LoadInnerXml(
   System.Xml.XmlNodeList nodeList
);
public void LoadInput(
   System.Object obj
);
```

Protected Methods

```
protected System.Xml.XmlNodeList GetInnerXml();
```

System.Security.Permissions Namespace (Visual J# Syntax)

Classes

CodeAccessSecurityAttribute

EnvironmentPermission

EnvironmentPermissionAttribute

FileDialogPermission

FileDialogPermissionAttribute

FileIOPermission

FileIOPermissionAttribute

IsolatedStorageFilePermission

IsolatedStorageFilePermissionAttribute

IsolatedStoragePermission

IsolatedStoragePermissionAttribute

PermissionSetAttribute

PrincipalPermission

PrincipalPermissionAttribute

PublisherIdentityPermission

PublisherIdentityPermissionAttribute

ReflectionPermission

ReflectionPermissionAttribute

RegistryPermission

RegistryPermissionAttribute

ResourcePermissionBase

ResourcePermissionBaseEntry

SecurityAttribute

SecurityPermission

SecurityPermissionAttribute

SiteIdentityPermission

SiteIdentityPermissionAttribute

StrongNameIdentityPermission

StrongNameIdentityPermissionAttribute

StrongNamePublicKeyBlob

UIPermission

UIPermissionAttribute

UrlIdentityPermission

UrlIdentityPermissionAttribute

ZoneIdentityPermission

ZoneIdentityPermissionAttribute

Interfaces

IUnrestrictedPermission

Enumerations

EnvironmentPermissionAccess

FileDialogPermissionAccess

FileIOPermissionAccess

IsolatedStorageContainment

PermissionState

ReflectionPermissionFlag

RegistryPermissionAccess

SecurityAction

SecurityPermissionFlag

UIPermissionClipboard

UIPermissionWindow

CodeAccessSecurityAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public abstract class CodeAccessSecurityAttribute
   extends System.Security.Permissions.SecurityAttribute
```

Public Constructors

```
public CodeAccessSecurityAttribute(
   System.Security.Permissions.SecurityAction action
);
```

EnvironmentPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class EnvironmentPermission
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Constructors

```
public EnvironmentPermission(
   System.Security.Permissions.PermissionState state
);
public EnvironmentPermission(
   System.Security.Permissions.EnvironmentPermissionAccess flag,
   System.String pathList
);
```

Public Methods

```
public void AddPathList(
   System.Security.Permissions.EnvironmentPermissionAccess flag,
   System.String pathList
);
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement esd
);
public System.String GetPathList(
   System.Security.Permissions.EnvironmentPermissionAccess flag
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public final boolean IsUnrestricted();
public void SetPathList(
   System.Security.Permissions.EnvironmentPermissionAccess flag,
   System.String pathList
);
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission other
);
```

EnvironmentPermissionAccess Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum EnvironmentPermissionAccess
```

Members

AllAccess

NoAccess

Read

Write

EnvironmentPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class EnvironmentPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public EnvironmentPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public void set_All(
   System.String value
);
public System.String get_Read();
public void set_Read(
   System.String value
);
public System.String get_Write();
public void set_Write(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

FileDialogPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class FileDialogPermission
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Constructors

```
public FileDialogPermission(
   System.Security.Permissions.FileDialogPermissionAccess access
);
public FileDialogPermission(
   System.Security.Permissions.PermissionState state
);
```

Public Properties

```
public System.Security.Permissions.FileDialogPermissionAccess get_Access();
public void set_Access(
   System.Security.Permissions.FileDialogPermissionAccess value
);
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement esd
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public final boolean IsUnrestricted();
public System.String ToString();
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

FileDialogPermissionAccess Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum FileDialogPermissionAccess
```

Members

None

Open

OpenSave

Save

FileDialogPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class FileDialogPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public FileDialogPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public boolean get_Open();
public void set_Open(
   boolean value
);
public boolean get_Save();
public void set_Save(
   boolean value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

FileIOPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class FileIOPermission
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Constructors

```
public FileIOPermission(
   System.Security.Permissions.PermissionState state
);
public FileIOPermission(
   System.Security.Permissions.FileIOPermissionAccess access,
   System.String path
);
public FileIOPermission(
   System.Security.Permissions.FileIOPermissionAccess access,
   System.String[] pathList
);
```

Public Properties

```
public System.Security.Permissions.FileIOPermissionAccess get_AllFiles();
public void set_AllFiles(
   System.Security.Permissions.FileIOPermissionAccess value
);
public System.Security.Permissions.FileIOPermissionAccess get_AllLocalFiles();
public void set_AllLocalFiles(
   System.Security.Permissions.FileIOPermissionAccess value
);
```

Public Methods

```
public void AddPathList(
   System.Security.Permissions.FileIOPermissionAccess access,
   System.String path
);
public void AddPathList(
   System.Security.Permissions.FileIOPermissionAccess access,
   System.String[] pathList
);
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement esd
);
public System.String[] GetPathList(
   System.Security.Permissions.FileIOPermissionAccess access
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
```

```
public final boolean IsUnrestricted();
public void SetPathList(
   System.Security.Permissions.FileIOPermissionAccess access,
   System.String path
);
public void SetPathList(
   System.Security.Permissions.FileIOPermissionAccess access,
   System.String[] pathList
);
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission other
);
```

FileIOPermissionAccess Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum FileIOPermissionAccess
```

Members

AllAccess

Append

NoAccess

PathDiscovery

Read

Write

FileIOPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class FileIOPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public FileIOPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public void set_All(
   System.String value
);
public System.String get_Append();
public void set_Append(
   System.String value
);
public System.String get_PathDiscovery();
public void set_PathDiscovery(
   System.String value
);
public System.String get_Read();
public void set_Read(
   System.String value
);
public System.String get_Write();
public void set_Write(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

IsolatedStorageContainment Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum IsolatedStorageContainment
```

Members

AdministerIsolatedStorageByUser

AssemblyIsolationByRoamingUser

AssemblyIsolationByUser

DomainIsolationByRoamingUser

DomainIsolationByUser

None

UnrestrictedIsolatedStorage

IsolatedStorageFilePermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class IsolatedStorageFilePermission
   extends System.Security.Permissions.IsolatedStoragePermission
```

Public Constructors

```
public IsolatedStorageFilePermission(
   System.Security.Permissions.PermissionState state
);
```

Public Methods

```
public System.Security.IPermission Copy();
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

IsolatedStorageFilePermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class IsolatedStorageFilePermissionAttribute
   extends System.Security.Permissions.IsolatedStoragePermissionAttribute
```

Public Constructors

```
public IsolatedStorageFilePermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

IsolatedStoragePermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class IsolatedStoragePermission
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Constructors

```
public IsolatedStoragePermission(
   System.Security.Permissions.PermissionState state
);
```

Public Properties

```
public System.Security.Permissions.IsolatedStorageContainment get_UsageAllowed();
public void set_UsageAllowed(
   System.Security.Permissions.IsolatedStorageContainment value
);
public long get_UserQuota();
public void set_UserQuota(
   long value
);
```

Public Methods

```
public void FromXml(
   System.Security.SecurityElement esd
);
public final boolean IsUnrestricted();
public System.Security.SecurityElement ToXml();
```

IsolatedStoragePermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public abstract class IsolatedStoragePermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public IsolatedStoragePermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.Security.Permissions.IsolatedStorageContainment get_UsageAllowed();
public void set_UsageAllowed(
   System.Security.Permissions.IsolatedStorageContainment value
);
public long get_UserQuota();
public void set_UserQuota(
   long value
);
```

IUnrestrictedPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IUnrestrictedPermission
```

Public Methods

```
boolean IsUnrestricted();
```

PermissionSetAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class PermissionSetAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public PermissionSetAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.String get_File();
public void set_File(
   System.String value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public boolean get_UnicodeEncoded();
public void set_UnicodeEncoded(
   boolean value
);
public System.String get_XML();
public void set_XML(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
public System.Security.PermissionSet CreatePermissionSet();
```

PermissionState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PermissionState
```

Members

None

Unrestricted

PrincipalPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class PrincipalPermission
   extends System.Object
   implements System.Security.IPermission, System.Security.ISecurityEncodable, System.Securit
y.Permissions.IUnrestrictedPermission, System.Security.Permissions.IBuiltInPermission
```

Public Constructors

```
public PrincipalPermission(
   System.Security.Permissions.PermissionState state
);
public PrincipalPermission(
   System.String name,
   System.String role
);
public PrincipalPermission(
   System.String name,
   System.String role,
   boolean isAuthenticated
);
```

Public Methods

```
public final System.Security.IPermission Copy();
public final void Demand();
public final void FromXml(
   System.Security.SecurityElement elem
);
public final System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public final boolean IsSubsetOf(
   System.Security.IPermission target
);
public final boolean IsUnrestricted();
public System.String ToString();
public final System.Security.SecurityElement ToXml();
public final System.Security.IPermission Union(
   System.Security.IPermission other
);
```

PrincipalPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Method, AllowMultiple = true,
Inherited = false) */
public class PrincipalPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public PrincipalPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public boolean get_Authenticated();
public void set_Authenticated(
   boolean value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_Role();
public void set_Role(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

PublisherIdentityPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class PublisherIdentityPermission
   extends System.Security.CodeAccessPermission
```

Public Constructors

```
public PublisherIdentityPermission(
   System.Security.Cryptography.X509Certificates.X509Certificate certificate
);
public PublisherIdentityPermission(
   System.Security.Permissions.PermissionState state
);
```

Public Properties

```
public System.Security.Cryptography.X509Certificates.X509Certificate get_Certificate();
public void set_Certificate(
   System.Security.Cryptography.X509Certificates.X509Certificate value
);
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement esd
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

PublisherIdentityPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class PublisherIdentityPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public PublisherIdentityPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.String get_CertFile();
public void set_CertFile(
   System.String value
);
public System.String get_SignedFile();
public void set_SignedFile(
   System.String value
);
public System.String get_X509Certificate();
public void set_X509Certificate(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

ReflectionPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ReflectionPermission
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Constructors

```
public ReflectionPermission(
   System.Security.Permissions.PermissionState state
);
public ReflectionPermission(
   System.Security.Permissions.ReflectionPermissionFlag flag
);
```

Public Properties

```
public System.Security.Permissions.ReflectionPermissionFlag get_Flags();
public void set_Flags(
   System.Security.Permissions.ReflectionPermissionFlag value
);
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement esd
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public final boolean IsUnrestricted();
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission other
);
```

ReflectionPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class ReflectionPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public ReflectionPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.Security.Permissions.ReflectionPermissionFlag get_Flags();
public void set_Flags(
   System.Security.Permissions.ReflectionPermissionFlag value
);
public boolean get_MemberAccess();
public void set_MemberAccess(
   boolean value
);
public boolean get_ReflectionEmit();
public void set_ReflectionEmit(
   boolean value
);
public boolean get_TypeInformation();
public void set_TypeInformation(
   boolean value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

ReflectionPermissionFlag Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum ReflectionPermissionFlag
```

Members

AllFlags

MemberAccess

NoFlags

ReflectionEmit

TypeInformation

RegistryPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RegistryPermission
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Constructors

```
public RegistryPermission(
   System.Security.Permissions.PermissionState state
);
public RegistryPermission(
   System.Security.Permissions.RegistryPermissionAccess access,
   System.String pathList
);
```

Public Methods

```
public void AddPathList(
   System.Security.Permissions.RegistryPermissionAccess access,
   System.String pathList
);
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement esd
);
public System.String GetPathList(
   System.Security.Permissions.RegistryPermissionAccess access
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public final boolean IsUnrestricted();
public void SetPathList(
   System.Security.Permissions.RegistryPermissionAccess access,
   System.String pathList
);
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission other
);
```

RegistryPermissionAccess Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum RegistryPermissionAccess
```

Members

AllAccess

Create

NoAccess

Read

Write

RegistryPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class RegistryPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public RegistryPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public void set_All(
   System.String value
);
public System.String get_Create();
public void set_Create(
   System.String value
);
public System.String get_Read();
public void set_Read(
   System.String value
);
public System.String get_Write();
public void set_Write(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

ResourcePermissionBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class ResourcePermissionBase
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement securityElement
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public final boolean IsUnrestricted();
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

Public Fields

```
public static System.String Any;
public static System.String Local;
```

Protected Constructors

```
protected ResourcePermissionBase();
protected ResourcePermissionBase(
   System.Security.Permissions.PermissionState state
);
```

Protected Properties

```
protected System.Type get_PermissionAccessType();
protected void set_PermissionAccessType(
   System.Type value
);
protected System.String[] get_TagNames();
protected void set_TagNames(
   System.String[] value
);
```

Protected Methods

```
protected void AddPermissionAccess(
```

```
   System.Security.Permissions.ResourcePermissionBaseEntry entry
);
protected void Clear();
protected System.Security.Permissions.ResourcePermissionBaseEntry[] GetPermissionEntries();
protected void RemovePermissionAccess(
   System.Security.Permissions.ResourcePermissionBaseEntry entry
);
```

ResourcePermissionBaseEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ResourcePermissionBaseEntry
   extends System.Object
```

Public Constructors

```
public ResourcePermissionBaseEntry();
public ResourcePermissionBaseEntry(
   int permissionAccess,
   System.String[] permissionAccessPath
);
```

Public Properties

```
public int get_PermissionAccess();
public System.String[] get_PermissionAccessPath();
```

SecurityAction Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SecurityAction
```

Members

Assert

Demand

Deny

InheritanceDemand

LinkDemand

PermitOnly

RequestMinimum

RequestOptional

RequestRefuse

SecurityAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public abstract class SecurityAttribute
   extends System.Attribute
```

Public Constructors

```
public SecurityAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.Security.Permissions.SecurityAction get_Action();
public void set_Action(
   System.Security.Permissions.SecurityAction value
);
public boolean get_Unrestricted();
public void set_Unrestricted(
   boolean value
);
```

Public Methods

```
public abstract System.Security.IPermission CreatePermission();
```

SecurityPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SecurityPermission
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Constructors

```
public SecurityPermission(
   System.Security.Permissions.PermissionState state
);
public SecurityPermission(
   System.Security.Permissions.SecurityPermissionFlag flag
);
```

Public Properties

```
public System.Security.Permissions.SecurityPermissionFlag get_Flags();
public void set_Flags(
   System.Security.Permissions.SecurityPermissionFlag value
);
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement esd
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public final boolean IsUnrestricted();
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

SecurityPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class SecurityPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public SecurityPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public boolean get_Assertion();
public void set_Assertion(
   boolean value
);
public boolean get_ControlAppDomain();
public void set_ControlAppDomain(
   boolean value
);
public boolean get_ControlDomainPolicy();
public void set_ControlDomainPolicy(
   boolean value
);
public boolean get_ControlEvidence();
public void set_ControlEvidence(
   boolean value
);
public boolean get_ControlPolicy();
public void set_ControlPolicy(
   boolean value
);
public boolean get_ControlPrincipal();
public void set_ControlPrincipal(
   boolean value
);
public boolean get_ControlThread();
public void set_ControlThread(
   boolean value
);
public boolean get_Execution();
public void set_Execution(
   boolean value
);
public System.Security.Permissions.SecurityPermissionFlag get_Flags();
public void set_Flags(
   System.Security.Permissions.SecurityPermissionFlag value
);
public boolean get_Infrastructure();
public void set_Infrastructure(
   boolean value
);
public boolean get_RemotingConfiguration();
```

```
public void set_RemotingConfiguration(
   boolean value
);
public boolean get_SerializationFormatter();
public void set_SerializationFormatter(
   boolean value
);
public boolean get_SkipVerification();
public void set_SkipVerification(
   boolean value
);
public boolean get_UnmanagedCode();
public void set_UnmanagedCode(
   boolean value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

SecurityPermissionFlag Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum SecurityPermissionFlag
```

Members

AllFlags

Assertion

ControlAppDomain

ControlDomainPolicy

ControlEvidence

ControlPolicy

ControlPrincipal

ControlThread

Execution

Infrastructure

NoFlags

RemotingConfiguration

SerializationFormatter

SkipVerification

UnmanagedCode

SiteIdentityPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SiteIdentityPermission
   extends System.Security.CodeAccessPermission
```

Public Constructors

```
public SiteIdentityPermission(
   System.Security.Permissions.PermissionState state
);
public SiteIdentityPermission(
   System.String site
);
```

Public Properties

```
public System.String get_Site();
public void set_Site(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement esd
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

SiteIdentityPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class SiteIdentityPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public SiteIdentityPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.String get_Site();
public void set_Site(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

StrongNameIdentityPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StrongNameIdentityPermission
   extends System.Security.CodeAccessPermission
```

Public Constructors

```
public StrongNameIdentityPermission(
   System.Security.Permissions.PermissionState state
);
public StrongNameIdentityPermission(
   System.Security.Permissions.StrongNamePublicKeyBlob blob,
   System.String name,
   System.Version version
);
```

Public Properties

```
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Security.Permissions.StrongNamePublicKeyBlob get_PublicKey();
public void set_PublicKey(
   System.Security.Permissions.StrongNamePublicKeyBlob value
);
public System.Version get_Version();
public void set_Version(
   System.Version value
);
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement e
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

StrongNameIdentityPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class StrongNameIdentityPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public StrongNameIdentityPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_PublicKey();
public void set_PublicKey(
   System.String value
);
public System.String get_Version();
public void set_Version(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

StrongNamePublicKeyBlob Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StrongNamePublicKeyBlob
   extends System.Object
```

Public Constructors

```
public StrongNamePublicKeyBlob(
   ubyte[] publicKey
);
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public System.String ToString();
```

UIPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class UIPermission
   extends System.Security.CodeAccessPermission
   implements System.Security.Permissions.IUnrestrictedPermission
```

Public Constructors

```
public UIPermission(
   System.Security.Permissions.PermissionState state
);
public UIPermission(
   System.Security.Permissions.UIPermissionClipboard clipboardFlag
);
public UIPermission(
   System.Security.Permissions.UIPermissionWindow windowFlag
);
public UIPermission(
   System.Security.Permissions.UIPermissionWindow windowFlag,
   System.Security.Permissions.UIPermissionClipboard clipboardFlag
);
```

Public Properties

```
public System.Security.Permissions.UIPermissionClipboard get_Clipboard();
public void set_Clipboard(
   System.Security.Permissions.UIPermissionClipboard value
);
public System.Security.Permissions.UIPermissionWindow get_Window();
public void set_Window(
   System.Security.Permissions.UIPermissionWindow value
);
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement esd
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public final boolean IsUnrestricted();
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

UIPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class UIPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public UIPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.Security.Permissions.UIPermissionClipboard get_Clipboard();
public void set_Clipboard(
   System.Security.Permissions.UIPermissionClipboard value
);
public System.Security.Permissions.UIPermissionWindow get_Window();
public void set_Window(
   System.Security.Permissions.UIPermissionWindow value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

UIPermissionClipboard Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum UIPermissionClipboard
```

Members

AllClipboard

NoClipboard

OwnClipboard

UIPermissionWindow Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum UIPermissionWindow
```

Members

AllWindows

NoWindows

SafeSubWindows

SafeTopLevelWindows

UrlIdentityPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class UrlIdentityPermission
   extends System.Security.CodeAccessPermission
```

Public Constructors

```
public UrlIdentityPermission(
   System.Security.Permissions.PermissionState state
);
public UrlIdentityPermission(
   System.String site
);
```

Public Properties

```
public System.String get_Url();
public void set_Url(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement esd
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

UrlIdentityPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class UrlIdentityPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public UrlIdentityPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.String get_Url();
public void set_Url(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

ZoneIdentityPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ZoneIdentityPermission
   extends System.Security.CodeAccessPermission
```

Public Constructors

```
public ZoneIdentityPermission(
   System.Security.Permissions.PermissionState state
);
public ZoneIdentityPermission(
   System.Security.SecurityZone zone
);
```

Public Properties

```
public System.Security.SecurityZone get_SecurityZone();
public void set_SecurityZone(
   System.Security.SecurityZone value
);
```

Public Methods

```
public System.Security.IPermission Copy();
public void FromXml(
   System.Security.SecurityElement esd
);
public System.Security.IPermission Intersect(
   System.Security.IPermission target
);
public boolean IsSubsetOf(
   System.Security.IPermission target
);
public System.Security.SecurityElement ToXml();
public System.Security.IPermission Union(
   System.Security.IPermission target
);
```

ZoneIdentityPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, AllowMultiple = true, Inherited = false) */
public class ZoneIdentityPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public ZoneIdentityPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.Security.SecurityZone get_Zone();
public void set_Zone(
   System.Security.SecurityZone value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

System.Security.Policy Namespace (Visual J# Syntax)

Classes

AllMembershipCondition

ApplicationDirectory

ApplicationDirectoryMembershipCondition

CodeGroup

Evidence

FileCodeGroup

FirstMatchCodeGroup

Hash

HashMembershipCondition

NetCodeGroup

PermissionRequestEvidence

PolicyException

PolicyLevel

PolicyStatement

Publisher

PublisherMembershipCondition

Site

SiteMembershipCondition

StrongName

StrongNameMembershipCondition

UnionCodeGroup

Url

UrlMembershipCondition

Zone

ZoneMembershipCondition

Interfaces

IIdentityPermissionFactory

IMembershipCondition

Enumerations

PolicyStatementAttribute

AllMembershipCondition Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class AllMembershipCondition
   extends System.Object
   implements System.Security.Policy.IMembershipCondition, System.Security.ISecurityEncodable
, System.Security.ISecurityPolicyEncodable, System.Security.Policy.IConstantMembershipConditi
on
```

Public Constructors

```
public AllMembershipCondition();
```

Public Methods

```
public final boolean Check(
   System.Security.Policy.Evidence evidence
);
public final System.Security.Policy.IMembershipCondition Copy();
public boolean Equals(
   System.Object o
);
public final void FromXml(
   System.Security.SecurityElement e
);
public final void FromXml(
   System.Security.SecurityElement e,
   System.Security.Policy.PolicyLevel level
);
public int GetHashCode();
public System.String ToString();
public final System.Security.SecurityElement ToXml();
public final System.Security.SecurityElement ToXml(
   System.Security.Policy.PolicyLevel level
);
```

ApplicationDirectory Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ApplicationDirectory
   extends System.Object
   implements System.Security.Policy.IBuiltInEvidence
```

Public Constructors

```
public ApplicationDirectory(
   System.String name
);
```

Public Properties

```
public System.String get_Directory();
```

Public Methods

```
public System.Object Copy();
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public System.String ToString();
```

ApplicationDirectoryMembershipCondition Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ApplicationDirectoryMembershipCondition
   extends System.Object
   implements System.Security.Policy.IMembershipCondition, System.Security.ISecurityEncodable
, System.Security.ISecurityPolicyEncodable, System.Security.Policy.IConstantMembershipConditi
on
```

Public Constructors

```
public ApplicationDirectoryMembershipCondition();
```

Public Methods

```
public final boolean Check(
   System.Security.Policy.Evidence evidence
);
public final System.Security.Policy.IMembershipCondition Copy();
public boolean Equals(
   System.Object o
);
public final void FromXml(
   System.Security.SecurityElement e
);
public final void FromXml(
   System.Security.SecurityElement e,
   System.Security.Policy.PolicyLevel level
);
public int GetHashCode();
public System.String ToString();
public final System.Security.SecurityElement ToXml();
public final System.Security.SecurityElement ToXml(
   System.Security.Policy.PolicyLevel level
);
```

CodeGroup Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class CodeGroup
   extends System.Object
```

Public Constructors

```
public CodeGroup(
   System.Security.Policy.IMembershipCondition membershipCondition,
   System.Security.Policy.PolicyStatement policy
);
```

Public Properties

```
public System.String get_AttributeString();
public System.Collections.IList get_Children();
public void set_Children(
   System.Collections.IList value
);
public System.String get_Description();
public void set_Description(
   System.String value
);
public System.Security.Policy.IMembershipCondition get_MembershipCondition();
public void set_MembershipCondition(
   System.Security.Policy.IMembershipCondition value
);
public abstract System.String get_MergeLogic();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_PermissionSetName();
public System.Security.Policy.PolicyStatement get_PolicyStatement();
public void set_PolicyStatement(
   System.Security.Policy.PolicyStatement value
);
```

Public Methods

```
public void AddChild(
   System.Security.Policy.CodeGroup group
);
public abstract System.Security.Policy.CodeGroup Copy();
public boolean Equals(
   System.Object o
);
public boolean Equals(
   System.Security.Policy.CodeGroup cg,
   boolean compareChildren
);
public void FromXml(
   System.Security.SecurityElement e
);
```

```
public void FromXml(
   System.Security.SecurityElement e,
   System.Security.Policy.PolicyLevel level
);
public int GetHashCode();
public void RemoveChild(
   System.Security.Policy.CodeGroup group
);
public abstract System.Security.Policy.PolicyStatement Resolve(
   System.Security.Policy.Evidence evidence
);
public abstract System.Security.Policy.CodeGroup ResolveMatchingCodeGroups(
   System.Security.Policy.Evidence evidence
);
public System.Security.SecurityElement ToXml();
public System.Security.SecurityElement ToXml(
   System.Security.Policy.PolicyLevel level
);
```

Protected Methods

```
protected void CreateXml(
   System.Security.SecurityElement element,
   System.Security.Policy.PolicyLevel level
);
protected void ParseXml(
   System.Security.SecurityElement e,
   System.Security.Policy.PolicyLevel level
);
```

Evidence Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Evidence
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public Evidence();
public Evidence(
   System.Security.Policy.Evidence evidence
);
public Evidence(
   System.Object[] hostEvidence,
   System.Object[] assemblyEvidence
);
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public boolean get_Locked();
public void set_Locked(
   boolean value
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public void AddAssembly(
   System.Object id
);
public void AddHost(
   System.Object id
);
public final void CopyTo(
   System.Array array,
   int index
);
public System.Collections.IEnumerator GetAssemblyEnumerator();
public final System.Collections.IEnumerator GetEnumerator();
public System.Collections.IEnumerator GetHostEnumerator();
public void Merge(
   System.Security.Policy.Evidence evidence
);
```

FileCodeGroup Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class FileCodeGroup
   extends System.Security.Policy.CodeGroup
```

Public Constructors

```
public FileCodeGroup(
   System.Security.Policy.IMembershipCondition membershipCondition,
   System.Security.Permissions.FileIOPermissionAccess access
);
```

Public Properties

```
public System.String get_AttributeString();
public System.String get_MergeLogic();
public System.String get_PermissionSetName();
```

Public Methods

```
public System.Security.Policy.CodeGroup Copy();
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public System.Security.Policy.PolicyStatement Resolve(
   System.Security.Policy.Evidence evidence
);
public System.Security.Policy.CodeGroup ResolveMatchingCodeGroups(
   System.Security.Policy.Evidence evidence
);
```

Protected Methods

```
protected void CreateXml(
   System.Security.SecurityElement element,
   System.Security.Policy.PolicyLevel level
);
protected void ParseXml(
   System.Security.SecurityElement e,
   System.Security.Policy.PolicyLevel level
);
```

FirstMatchCodeGroup Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class FirstMatchCodeGroup
   extends System.Security.Policy.CodeGroup
```

Public Constructors

```
public FirstMatchCodeGroup(
   System.Security.Policy.IMembershipCondition membershipCondition,
   System.Security.Policy.PolicyStatement policy
);
```

Public Properties

```
public System.String get_MergeLogic();
```

Public Methods

```
public System.Security.Policy.CodeGroup Copy();
public System.Security.Policy.PolicyStatement Resolve(
   System.Security.Policy.Evidence evidence
);
public System.Security.Policy.CodeGroup ResolveMatchingCodeGroups(
   System.Security.Policy.Evidence evidence
);
```

Hash Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Hash
   extends System.Object
   implements System.Runtime.Serialization.ISerializable, System.Security.Policy.IBuiltInEvid
ence
```

Public Constructors

```
public Hash(
   System.Reflection.Assembly assembly
);
```

Public Properties

```
public ubyte[] get_MD5();
public ubyte[] get_SHA1();
```

Public Methods

```
public ubyte[] GenerateHash(
   System.Security.Cryptography.HashAlgorithm hashAlg
);
public final void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
public System.String ToString();
```

HashMembershipCondition Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class HashMembershipCondition
   extends System.Object
   implements System.Security.Policy.IMembershipCondition, System.Security.ISecurityEncodable
, System.Security.ISecurityPolicyEncodable
```

Public Constructors

```
public HashMembershipCondition(
   System.Security.Cryptography.HashAlgorithm hashAlg,
   ubyte[] value
);
```

Public Properties

```
public System.Security.Cryptography.HashAlgorithm get_HashAlgorithm();
public void set_HashAlgorithm(
   System.Security.Cryptography.HashAlgorithm value
);
public ubyte[] get_HashValue();
public void set_HashValue(
   ubyte[] value
);
```

Public Methods

```
public final boolean Check(
   System.Security.Policy.Evidence evidence
);
public final System.Security.Policy.IMembershipCondition Copy();
public boolean Equals(
   System.Object o
);
public final void FromXml(
   System.Security.SecurityElement e
);
public final void FromXml(
   System.Security.SecurityElement e,
   System.Security.Policy.PolicyLevel level
);
public int GetHashCode();
public System.String ToString();
public final System.Security.SecurityElement ToXml();
public final System.Security.SecurityElement ToXml(
   System.Security.Policy.PolicyLevel level
);
```

IIdentityPermissionFactory Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IIdentityPermissionFactory
```

Public Methods

```
System.Security.IPermission CreateIdentityPermission(
   System.Security.Policy.Evidence evidence
);
```

IMembershipCondition Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IMembershipCondition
   implements System.Security.ISecurityEncodable, System.Security.ISecurityPolicyEncodable
```

Public Methods

```
boolean Check(
   System.Security.Policy.Evidence evidence
);
System.Security.Policy.IMembershipCondition Copy();
boolean Equals(
   System.Object obj
);
System.String ToString();
```

NetCodeGroup Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class NetCodeGroup
   extends System.Security.Policy.CodeGroup
```

Public Constructors

```
public NetCodeGroup(
   System.Security.Policy.IMembershipCondition membershipCondition
);
```

Public Properties

```
public System.String get_AttributeString();
public System.String get_MergeLogic();
public System.String get_PermissionSetName();
```

Public Methods

```
public System.Security.Policy.CodeGroup Copy();
public System.Security.Policy.PolicyStatement Resolve(
   System.Security.Policy.Evidence evidence
);
public System.Security.Policy.CodeGroup ResolveMatchingCodeGroups(
   System.Security.Policy.Evidence evidence
);
```

PermissionRequestEvidence Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class PermissionRequestEvidence
   extends System.Object
   implements System.Security.Policy.IBuiltInEvidence
```

Public Constructors

```
public PermissionRequestEvidence(
   System.Security.PermissionSet request,
   System.Security.PermissionSet optional,
   System.Security.PermissionSet denied
);
```

Public Properties

```
public System.Security.PermissionSet get_DeniedPermissions();
public System.Security.PermissionSet get_OptionalPermissions();
public System.Security.PermissionSet get_RequestedPermissions();
```

Public Methods

```
public System.Security.Policy.PermissionRequestEvidence Copy();
public System.String ToString();
```

PolicyException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class PolicyException
   extends System.SystemException
```

Public Constructors

```
public PolicyException();
public PolicyException(
   System.String message
);
public PolicyException(
   System.String message,
   System.Exception exception
);
```

Protected Constructors

```
protected PolicyException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

PolicyLevel Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class PolicyLevel
   extends System.Object
```

Public Properties

```
public System.Collections.IList get_FullTrustAssemblies();
public System.String get_Label();
public System.Collections.IList get_NamedPermissionSets();
public System.Security.Policy.CodeGroup get_RootCodeGroup();
public void set_RootCodeGroup(
   System.Security.Policy.CodeGroup value
);
public System.String get_StoreLocation();
```

Public Methods

```
public void AddFullTrustAssembly(
   System.Security.Policy.StrongName sn
);
public void AddFullTrustAssembly(
   System.Security.Policy.StrongNameMembershipCondition snMC
);
public void AddNamedPermissionSet(
   System.Security.NamedPermissionSet permSet
);
public System.Security.NamedPermissionSet ChangeNamedPermissionSet(
   System.String name,
   System.Security.PermissionSet pSet
);
public static System.Security.Policy.PolicyLevel CreateAppDomainLevel();
public void FromXml(
   System.Security.SecurityElement e
);
public System.Security.NamedPermissionSet GetNamedPermissionSet(
   System.String name
);
public void Recover();
public void RemoveFullTrustAssembly(
   System.Security.Policy.StrongName sn
);
public void RemoveFullTrustAssembly(
   System.Security.Policy.StrongNameMembershipCondition snMC
);
public System.Security.NamedPermissionSet RemoveNamedPermissionSet(
   System.Security.NamedPermissionSet permSet
);
public System.Security.NamedPermissionSet RemoveNamedPermissionSet(
   System.String name
);
public void Reset();
public System.Security.Policy.PolicyStatement Resolve(
   System.Security.Policy.Evidence evidence
);
public System.Security.Policy.CodeGroup ResolveMatchingCodeGroups(
```

```
    System.Security.Policy.Evidence evidence
);
public System.Security.SecurityElement ToXml();
```

PolicyStatement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class PolicyStatement
   extends System.Object
   implements System.Security.ISecurityEncodable, System.Security.ISecurityPolicyEncodable
```

Public Constructors

```
public PolicyStatement(
   System.Security.PermissionSet permSet
);
public PolicyStatement(
   System.Security.PermissionSet permSet,
   System.Security.Policy.PolicyStatementAttribute attributes
);
```

Public Properties

```
public System.Security.Policy.PolicyStatementAttribute get_Attributes();
public void set_Attributes(
   System.Security.Policy.PolicyStatementAttribute value
);
public System.String get_AttributeString();
public System.Security.PermissionSet get_PermissionSet();
public void set_PermissionSet(
   System.Security.PermissionSet value
);
```

Public Methods

```
public System.Security.Policy.PolicyStatement Copy();
public final void FromXml(
   System.Security.SecurityElement et
);
public final void FromXml(
   System.Security.SecurityElement et,
   System.Security.Policy.PolicyLevel level
);
public final System.Security.SecurityElement ToXml();
public final System.Security.SecurityElement ToXml(
   System.Security.Policy.PolicyLevel level
);
```

PolicyStatementAttribute Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum PolicyStatementAttribute
```

Members

All

Exclusive

LevelFinal

Nothing

Publisher Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Publisher
   extends System.Object
   implements System.Security.Policy.IIdentityPermissionFactory, System.Security.Policy.IBuil
tInEvidence
```

Public Constructors

```
public Publisher(
   System.Security.Cryptography.X509Certificates.X509Certificate cert
);
```

Public Properties

```
public System.Security.Cryptography.X509Certificates.X509Certificate get_Certificate();
```

Public Methods

```
public System.Object Copy();
public final System.Security.IPermission CreateIdentityPermission(
   System.Security.Policy.Evidence evidence
);
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public System.String ToString();
```

PublisherMembershipCondition Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class PublisherMembershipCondition
   extends System.Object
   implements System.Security.Policy.IMembershipCondition, System.Security.ISecurityEncodable
, System.Security.ISecurityPolicyEncodable, System.Security.Policy.IConstantMembershipConditi
on
```

Public Constructors

```
public PublisherMembershipCondition(
   System.Security.Cryptography.X509Certificates.X509Certificate certificate
);
```

Public Properties

```
public System.Security.Cryptography.X509Certificates.X509Certificate get_Certificate();
public void set_Certificate(
   System.Security.Cryptography.X509Certificates.X509Certificate value
);
```

Public Methods

```
public final boolean Check(
   System.Security.Policy.Evidence evidence
);
public final System.Security.Policy.IMembershipCondition Copy();
public boolean Equals(
   System.Object o
);
public final void FromXml(
   System.Security.SecurityElement e
);
public final void FromXml(
   System.Security.SecurityElement e,
   System.Security.Policy.PolicyLevel level
);
public int GetHashCode();
public System.String ToString();
public final System.Security.SecurityElement ToXml();
public final System.Security.SecurityElement ToXml(
   System.Security.Policy.PolicyLevel level
);
```

Site Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Site
   extends System.Object
   implements System.Security.Policy.IIdentityPermissionFactory, System.Security.Policy.IBuil
tInEvidence
```

Public Constructors

```
public Site(
   System.String name
);
```

Public Properties

```
public System.String get_Name();
```

Public Methods

```
public System.Object Copy();
public static System.Security.Policy.Site CreateFromUrl(
   System.String url
);
public final System.Security.IPermission CreateIdentityPermission(
   System.Security.Policy.Evidence evidence
);
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public System.String ToString();
```

SiteMembershipCondition Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SiteMembershipCondition
   extends System.Object
   implements System.Security.Policy.IMembershipCondition, System.Security.ISecurityEncodable
, System.Security.ISecurityPolicyEncodable, System.Security.Policy.IConstantMembershipConditi
on
```

Public Constructors

```
public SiteMembershipCondition(
   System.String site
);
```

Public Properties

```
public System.String get_Site();
public void set_Site(
   System.String value
);
```

Public Methods

```
public final boolean Check(
   System.Security.Policy.Evidence evidence
);
public final System.Security.Policy.IMembershipCondition Copy();
public boolean Equals(
   System.Object o
);
public final void FromXml(
   System.Security.SecurityElement e
);
public final void FromXml(
   System.Security.SecurityElement e,
   System.Security.Policy.PolicyLevel level
);
public int GetHashCode();
public System.String ToString();
public final System.Security.SecurityElement ToXml();
public final System.Security.SecurityElement ToXml(
   System.Security.Policy.PolicyLevel level
);
```

StrongName Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StrongName
   extends System.Object
   implements System.Security.Policy.IIdentityPermissionFactory, System.Security.Policy.IBuil
tInEvidence
```

Public Constructors

```
public StrongName(
   System.Security.Permissions.StrongNamePublicKeyBlob blob,
   System.String name,
   System.Version version
);
```

Public Properties

```
public System.String get_Name();
public System.Security.Permissions.StrongNamePublicKeyBlob get_PublicKey();
public System.Version get_Version();
```

Public Methods

```
public System.Object Copy();
public final System.Security.IPermission CreateIdentityPermission(
   System.Security.Policy.Evidence evidence
);
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public System.String ToString();
```

StrongNameMembershipCondition Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class StrongNameMembershipCondition
   extends System.Object
   implements System.Security.Policy.IMembershipCondition, System.Security.ISecurityEncodable
, System.Security.ISecurityPolicyEncodable, System.Security.Policy.IConstantMembershipConditi
on
```

Public Constructors

```
public StrongNameMembershipCondition(
   System.Security.Permissions.StrongNamePublicKeyBlob blob,
   System.String name,
   System.Version version
);
```

Public Properties

```
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Security.Permissions.StrongNamePublicKeyBlob get_PublicKey();
public void set_PublicKey(
   System.Security.Permissions.StrongNamePublicKeyBlob value
);
public System.Version get_Version();
public void set_Version(
   System.Version value
);
```

Public Methods

```
public final boolean Check(
   System.Security.Policy.Evidence evidence
);
public final System.Security.Policy.IMembershipCondition Copy();
public boolean Equals(
   System.Object o
);
public final void FromXml(
   System.Security.SecurityElement e
);
public final void FromXml(
   System.Security.SecurityElement e,
   System.Security.Policy.PolicyLevel level
);
public int GetHashCode();
public System.String ToString();
public final System.Security.SecurityElement ToXml();
public final System.Security.SecurityElement ToXml(
   System.Security.Policy.PolicyLevel level
);
```

UnionCodeGroup Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class UnionCodeGroup
   extends System.Security.Policy.CodeGroup
```

Public Constructors

```
public UnionCodeGroup(
   System.Security.Policy.IMembershipCondition membershipCondition,
   System.Security.Policy.PolicyStatement policy
);
```

Public Properties

```
public System.String get_MergeLogic();
```

Public Methods

```
public System.Security.Policy.CodeGroup Copy();
public System.Security.Policy.PolicyStatement Resolve(
   System.Security.Policy.Evidence evidence
);
public System.Security.Policy.CodeGroup ResolveMatchingCodeGroups(
   System.Security.Policy.Evidence evidence
);
```

Url Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Url
   extends System.Object
   implements System.Security.Policy.IIdentityPermissionFactory, System.Security.Policy.IBuil
tInEvidence
```

Public Constructors

```
public Url(
   System.String name
);
```

Public Properties

```
public System.String get_Value();
```

Public Methods

```
public System.Object Copy();
public final System.Security.IPermission CreateIdentityPermission(
   System.Security.Policy.Evidence evidence
);
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public System.String ToString();
```

UrlMembershipCondition Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class UrlMembershipCondition
   extends System.Object
   implements System.Security.Policy.IMembershipCondition, System.Security.ISecurityEncodable
, System.Security.ISecurityPolicyEncodable, System.Security.Policy.IConstantMembershipConditi
on
```

Public Constructors

```
public UrlMembershipCondition(
   System.String url
);
```

Public Properties

```
public System.String get_Url();
public void set_Url(
   System.String value
);
```

Public Methods

```
public final boolean Check(
   System.Security.Policy.Evidence evidence
);
public final System.Security.Policy.IMembershipCondition Copy();
public boolean Equals(
   System.Object o
);
public final void FromXml(
   System.Security.SecurityElement e
);
public final void FromXml(
   System.Security.SecurityElement e,
   System.Security.Policy.PolicyLevel level
);
public int GetHashCode();
public System.String ToString();
public final System.Security.SecurityElement ToXml();
public final System.Security.SecurityElement ToXml(
   System.Security.Policy.PolicyLevel level
);
```

Zone Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Zone
   extends System.Object
   implements System.Security.Policy.IIdentityPermissionFactory, System.Security.Policy.IBuil
tInEvidence
```

Public Constructors

```
public Zone(
   System.Security.SecurityZone zone
);
```

Public Properties

```
public System.Security.SecurityZone get_SecurityZone();
```

Public Methods

```
public System.Object Copy();
public static System.Security.Policy.Zone CreateFromUrl(
   System.String url
);
public final System.Security.IPermission CreateIdentityPermission(
   System.Security.Policy.Evidence evidence
);
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public System.String ToString();
```

ZoneMembershipCondition Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ZoneMembershipCondition
   extends System.Object
   implements System.Security.Policy.IMembershipCondition, System.Security.ISecurityEncodable
, System.Security.ISecurityPolicyEncodable, System.Security.Policy.IConstantMembershipConditi
on
```

Public Constructors

```
public ZoneMembershipCondition(
   System.Security.SecurityZone zone
);
```

Public Properties

```
public System.Security.SecurityZone get_SecurityZone();
public void set_SecurityZone(
   System.Security.SecurityZone value
);
```

Public Methods

```
public final boolean Check(
   System.Security.Policy.Evidence evidence
);
public final System.Security.Policy.IMembershipCondition Copy();
public boolean Equals(
   System.Object o
);
public final void FromXml(
   System.Security.SecurityElement e
);
public final void FromXml(
   System.Security.SecurityElement e,
   System.Security.Policy.PolicyLevel level
);
public int GetHashCode();
public System.String ToString();
public final System.Security.SecurityElement ToXml();
public final System.Security.SecurityElement ToXml(
   System.Security.Policy.PolicyLevel level
);
```

System.Security.Principal Namespace (Visual J# Syntax)

Classes

GenericIdentity

GenericPrincipal

WindowsIdentity

WindowsImpersonationContext

WindowsPrincipal

Interfaces

IIdentity

IPrincipal

Enumerations

PrincipalPolicy

WindowsAccountType

WindowsBuiltInRole

GenericIdentity Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class GenericIdentity
   extends System.Object
   implements System.Security.Principal.IIdentity
```

Public Constructors

```
public GenericIdentity(
   System.String name
);
public GenericIdentity(
   System.String name,
   System.String type
);
```

Public Properties

```
public System.String get_AuthenticationType();
public boolean get_IsAuthenticated();
public System.String get_Name();
```

GenericPrincipal Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class GenericPrincipal
   extends System.Object
   implements System.Security.Principal.IPrincipal
```

Public Constructors

```
public GenericPrincipal(
   System.Security.Principal.IIdentity identity,
   System.String[] roles
);
```

Public Properties

```
public System.Security.Principal.IIdentity get_Identity();
```

Public Methods

```
public boolean IsInRole(
   System.String role
);
```

IIdentity Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IIdentity
```

Public Properties

```
System.String get_AuthenticationType();
boolean get_IsAuthenticated();
System.String get_Name();
```

IPrincipal Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IPrincipal
```

Public Properties

```
System.Security.Principal.IIdentity get_Identity();
```

Public Methods

```
boolean IsInRole(
   System.String role
);
```

PrincipalPolicy Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PrincipalPolicy
```

Members

NoPrincipal

UnauthenticatedPrincipal

WindowsPrincipal

WindowsAccountType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum WindowsAccountType
```

Members

Anonymous

Guest

Normal

System

WindowsBuiltInRole Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum WindowsBuiltInRole
```

Members

AccountOperator

Administrator

BackupOperator

Guest

PowerUser

PrintOperator

Replicator

SystemOperator

User

WindowsIdentity Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class WindowsIdentity
   extends System.Object
   implements System.Security.Principal.IIdentity, System.Runtime.Serialization.IDeserializat
ionCallback
```

Public Constructors

```
public WindowsIdentity(
   System.IntPtr userToken
);
public WindowsIdentity(
   System.IntPtr userToken,
   System.String type
);
public WindowsIdentity(
   System.IntPtr userToken,
   System.String type,
   System.Security.Principal.WindowsAccountType acctType
);
public WindowsIdentity(
   System.IntPtr userToken,
   System.String type,
   System.Security.Principal.WindowsAccountType acctType,
   boolean isAuthenticated
);
```

Public Properties

```
public System.String get_AuthenticationType();
public boolean get_IsAnonymous();
public boolean get_IsAuthenticated();
public boolean get_IsGuest();
public boolean get_IsSystem();
public System.String get_Name();
public System.IntPtr get_Token();
```

Public Methods

```
public static System.Security.Principal.WindowsIdentity GetAnonymous();
public static System.Security.Principal.WindowsIdentity GetCurrent();
public System.Security.Principal.WindowsImpersonationContext Impersonate();
public static System.Security.Principal.WindowsImpersonationContext Impersonate(
   System.IntPtr userToken
);
```

Protected Methods

```
protected void Finalize();
```

WindowsImpersonationContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WindowsImpersonationContext
   extends System.Object
```

Public Methods

```
public void Undo();
```

Protected Methods

```
protected void Finalize();
```

WindowsPrincipal Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class WindowsPrincipal
   extends System.Object
   implements System.Security.Principal.IPrincipal
```

Public Constructors

```
public WindowsPrincipal(
   System.Security.Principal.WindowsIdentity ntIdentity
);
```

Public Properties

```
public System.Security.Principal.IIdentity get_Identity();
```

Public Methods

```
public boolean IsInRole(
   int rid
);
public boolean IsInRole(
   System.Security.Principal.WindowsBuiltInRole role
);
public boolean IsInRole(
   System.String role
);
```

System.ServiceProcess Namespace (Visual J# Syntax)

Classes

ServiceBase

ServiceController

ServiceControllerPermission

ServiceControllerPermissionAttribute

ServiceControllerPermissionEntry

ServiceControllerPermissionEntryCollection

ServiceInstaller

ServiceProcessDescriptionAttribute

ServiceProcessInstaller

TimeoutException

Enumerations

PowerBroadcastStatus

ServiceAccount

ServiceControllerPermissionAccess

ServiceControllerStatus

ServiceStartMode

ServiceType

PowerBroadcastStatus Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PowerBroadcastStatus
```

Members

BatteryLow

OemEvent

PowerStatusChange

QuerySuspend

QuerySuspendFailed

ResumeAutomatic

ResumeCritical

ResumeSuspend

Suspend

ServiceAccount Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ServiceAccount
```

Members

LocalService

LocalSystem

NetworkService

User

ServiceBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IRootDesigner", DesignerTypeName = "Microsoft.VisualStudio.Install.UserNTServi
ceDesigner, Microsoft.VisualStudio, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5
f7f11d50a3a") */
/** @attribute System.ComponentModel.InstallerTypeAttribute(InstallerType = System.ServicePro
cess.ServiceProcessInstaller) */
public class ServiceBase
   extends System.ComponentModel.Component
```

Public Constructors

```
public ServiceBase();
```

Public Properties

```
public boolean get_AutoLog();
public void set_AutoLog(
   boolean value
);
public boolean get_CanHandlePowerEvent();
public void set_CanHandlePowerEvent(
   boolean value
);
public boolean get_CanPauseAndContinue();
public void set_CanPauseAndContinue(
   boolean value
);
public boolean get_CanShutdown();
public void set_CanShutdown(
   boolean value
);
public boolean get_CanStop();
public void set_CanStop(
   boolean value
);
public System.Diagnostics.EventLog get_EventLog();
public System.String get_ServiceName();
public void set_ServiceName(
   System.String value
);
```

Public Methods

```
public static void Run(
   System.ServiceProcess.ServiceBase service
);
public static void Run(
   System.ServiceProcess.ServiceBase[] services
);
```

Public Fields

```
public static int MaxNameLength;
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void OnContinue();
protected void OnCustomCommand(
   int command
);
protected void OnPause();
protected boolean OnPowerEvent(
   System.ServiceProcess.PowerBroadcastStatus powerStatus
);
protected void OnShutdown();
protected void OnStart(
   System.String[] args
);
protected void OnStop();
```

ServiceController Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.ServiceProcess.Design.ServiceController
Designer, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
") */
public class ServiceController
   extends System.ComponentModel.Component
```

Public Constructors

```
public ServiceController();
public ServiceController(
   System.String name
);
public ServiceController(
   System.String name,
   System.String machineName
);
```

Public Properties

```
public boolean get_CanPauseAndContinue();
public boolean get_CanShutdown();
public boolean get_CanStop();
public System.ServiceProcess.ServiceController[] get_DependentServices();
public System.String get_DisplayName();
public void set_DisplayName(
   System.String value
);
public System.String get_MachineName();
public void set_MachineName(
   System.String value
);
public System.String get_ServiceName();
public void set_ServiceName(
   System.String value
);
public System.ServiceProcess.ServiceController[] get_ServicesDependedOn();
public System.ServiceProcess.ServiceType get_ServiceType();
public System.ServiceProcess.ServiceControllerStatus get_Status();
```

Public Methods

```
public void Close();
public void Continue();
public void ExecuteCommand(
   int command
);
public static System.ServiceProcess.ServiceController[] GetDevices();
public static System.ServiceProcess.ServiceController[] GetDevices(
   System.String machineName
);
public static System.ServiceProcess.ServiceController[] GetServices();
public static System.ServiceProcess.ServiceController[] GetServices(
```

```
   System.String machineName
);
public void Pause();
public void Refresh();
public void Start();
public void Start(
   System.String[] args
);
public void Stop();
public void WaitForStatus(
   System.ServiceProcess.ServiceControllerStatus desiredStatus
);
public void WaitForStatus(
   System.ServiceProcess.ServiceControllerStatus desiredStatus,
   System.TimeSpan timeout
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

ServiceControllerPermission Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ServiceControllerPermission
   extends System.Security.Permissions.ResourcePermissionBase
```

Public Constructors

```
public ServiceControllerPermission();
public ServiceControllerPermission(
   System.Security.Permissions.PermissionState state
);
public ServiceControllerPermission(
   System.ServiceProcess.ServiceControllerPermissionEntry[] permissionAccessEntries
);
public ServiceControllerPermission(
   System.ServiceProcess.ServiceControllerPermissionAccess permissionAccess,
   System.String machineName,
   System.String serviceName
);
```

Public Properties

```
public System.ServiceProcess.ServiceControllerPermissionEntryCollection get_PermissionEntries
();
```

ServiceControllerPermissionAccess Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum ServiceControllerPermissionAccess
```

Members

Browse

Control

None

ServiceControllerPermissionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, Class, Struct, Constructor,
Method, Event, AllowMultiple = true, Inherited = false) */
public class ServiceControllerPermissionAttribute
   extends System.Security.Permissions.CodeAccessSecurityAttribute
```

Public Constructors

```
public ServiceControllerPermissionAttribute(
   System.Security.Permissions.SecurityAction action
);
```

Public Properties

```
public System.String get_MachineName();
public void set_MachineName(
   System.String value
);
public System.ServiceProcess.ServiceControllerPermissionAccess get_PermissionAccess();
public void set_PermissionAccess(
   System.ServiceProcess.ServiceControllerPermissionAccess value
);
public System.String get_ServiceName();
public void set_ServiceName(
   System.String value
);
```

Public Methods

```
public System.Security.IPermission CreatePermission();
```

ServiceControllerPermissionEntry Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ServiceControllerPermissionEntry
   extends System.Object
```

Public Constructors

```
public ServiceControllerPermissionEntry();
public ServiceControllerPermissionEntry(
   System.ServiceProcess.ServiceControllerPermissionAccess permissionAccess,
   System.String machineName,
   System.String serviceName
);
```

Public Properties

```
public System.String get_MachineName();
public System.ServiceProcess.ServiceControllerPermissionAccess get_PermissionAccess();
public System.String get_ServiceName();
```

ServiceControllerPermissionEntryCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ServiceControllerPermissionEntryCollection
   extends System.Collections.CollectionBase
```

Public Properties

```
public System.ServiceProcess.ServiceControllerPermissionEntry get_Item(
   int index
);
public void set_Item(
   int index,
   System.ServiceProcess.ServiceControllerPermissionEntry value
);
```

Public Methods

```
public int Add(
   System.ServiceProcess.ServiceControllerPermissionEntry value
);
public void AddRange(
   System.ServiceProcess.ServiceControllerPermissionEntryCollection value
);
public void AddRange(
   System.ServiceProcess.ServiceControllerPermissionEntry[] value
);
public boolean Contains(
   System.ServiceProcess.ServiceControllerPermissionEntry value
);
public void CopyTo(
   System.ServiceProcess.ServiceControllerPermissionEntry[] array,
   int index
);
public int IndexOf(
   System.ServiceProcess.ServiceControllerPermissionEntry value
);
public void Insert(
   int index,
   System.ServiceProcess.ServiceControllerPermissionEntry value
);
public void Remove(
   System.ServiceProcess.ServiceControllerPermissionEntry value
);
```

Protected Methods

```
protected void OnClear();
protected void OnInsert(
   int index,
   System.Object value
```

```
);
protected void OnRemove(
   int index,
   System.Object value
);
protected void OnSet(
   int index,
   System.Object oldValue,
   System.Object newValue
);
```

ServiceControllerStatus Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ServiceControllerStatus
```

Members

ContinuePending

Paused

PausePending

Running

StartPending

Stopped

StopPending

ServiceInstaller Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ServiceInstaller
   extends System.Configuration.Install.ComponentInstaller
```

Public Constructors

```
public ServiceInstaller();
```

Public Properties

```
public System.String get_DisplayName();
public void set_DisplayName(
   System.String value
);
public System.String get_ServiceName();
public void set_ServiceName(
   System.String value
);
public System.String[] get_ServicesDependedOn();
public void set_ServicesDependedOn(
   System.String[] value
);
public System.ServiceProcess.ServiceStartMode get_StartType();
public void set_StartType(
   System.ServiceProcess.ServiceStartMode value
);
```

Public Methods

```
public void CopyFromComponent(
   System.ComponentModel.IComponent component
);
public void Install(
   System.Collections.IDictionary stateSaver
);
public boolean IsEquivalentInstaller(
   System.Configuration.Install.ComponentInstaller otherInstaller
);
public void Rollback(
   System.Collections.IDictionary savedState
);
public void Uninstall(
   System.Collections.IDictionary savedState
);
```

ServiceProcessDescriptionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class ServiceProcessDescriptionAttribute
   extends System.ComponentModel.DescriptionAttribute
```

Public Constructors

```
public ServiceProcessDescriptionAttribute(
   System.String description
);
```

Public Properties

```
public System.String get_Description();
```

ServiceProcessInstaller Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ServiceProcessInstaller
   extends System.Configuration.Install.ComponentInstaller
```

Public Constructors

```
public ServiceProcessInstaller();
```

Public Properties

```
public System.ServiceProcess.ServiceAccount get_Account();
public void set_Account(
   System.ServiceProcess.ServiceAccount value
);
public System.String get_HelpText();
public System.String get_Password();
public void set_Password(
   System.String value
);
public System.String get_Username();
public void set_Username(
   System.String value
);
```

Public Methods

```
public void CopyFromComponent(
   System.ComponentModel.IComponent comp
);
public void Install(
   System.Collections.IDictionary stateSaver
);
public void Rollback(
   System.Collections.IDictionary savedState
);
```

ServiceStartMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ServiceStartMode
```

Members

Automatic

Disabled

Manual

ServiceType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum ServiceType
```

Members

Adapter

FileSystemDriver

InteractiveProcess

KernelDriver

RecognizerDriver

Win32OwnProcess

Win32ShareProcess

TimeoutException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class TimeoutException
   extends System.SystemException
```

Public Constructors

```
public TimeoutException();
public TimeoutException(
   System.String message
);
```

Protected Constructors

```
protected TimeoutException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

System.Text Namespace (Visual J# Syntax)

Classes

ASCIIEncoding

Decoder

Encoder

Encoding

StringBuilder

UnicodeEncoding

UTF7Encoding

UTF8Encoding

ASCIIEncoding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ASCIIEncoding
   extends System.Text.Encoding
```

Public Constructors

```
public ASCIIEncoding();
```

Public Methods

```
public int GetByteCount(
   System.String chars
);
public int GetByteCount(
   char[] chars,
   int index,
   int count
);
public int GetBytes(
   char[] chars,
   int charIndex,
   int charCount,
   ubyte[] bytes,
   int byteIndex
);
public int GetBytes(
   System.String chars,
   int charIndex,
   int charCount,
   ubyte[] bytes,
   int byteIndex
);
public int GetCharCount(
   ubyte[] bytes,
   int index,
   int count
);
public int GetChars(
   ubyte[] bytes,
   int byteIndex,
   int byteCount,
   char[] chars,
   int charIndex
);
public int GetMaxByteCount(
   int charCount
);
public int GetMaxCharCount(
   int byteCount
);
public System.String GetString(
   ubyte[] bytes
);
public System.String GetString(
   ubyte[] bytes,
```

```
    int byteIndex,
    int byteCount
);
```

Decoder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class Decoder
   extends System.Object
```

Public Methods

```
public abstract int GetCharCount(
   ubyte[] bytes,
   int index,
   int count
);
public abstract int GetChars(
   ubyte[] bytes,
   int byteIndex,
   int byteCount,
   char[] chars,
   int charIndex
);
```

Protected Constructors

```
protected Decoder();
```

Encoder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class Encoder
   extends System.Object
```

Public Methods

```
public abstract int GetByteCount(
   char[] chars,
   int index,
   int count,
   boolean flush
);
public abstract int GetBytes(
   char[] chars,
   int charIndex,
   int charCount,
   ubyte[] bytes,
   int byteIndex,
   boolean flush
);
```

Protected Constructors

```
protected Encoder();
```

Encoding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public abstract class Encoding
   extends System.Object
```

Public Properties

```
public static System.Text.Encoding get_ASCII();
public static System.Text.Encoding get_BigEndianUnicode();
public System.String get_BodyName();
public int get_CodePage();
public static System.Text.Encoding get_Default();
public System.String get_EncodingName();
public System.String get_HeaderName();
public boolean get_IsBrowserDisplay();
public boolean get_IsBrowserSave();
public boolean get_IsMailNewsDisplay();
public boolean get_IsMailNewsSave();
public static System.Text.Encoding get_Unicode();
public static System.Text.Encoding get_UTF7();
public static System.Text.Encoding get_UTF8();
public System.String get_WebName();
public int get_WindowsCodePage();
```

Public Methods

```
public static ubyte[] Convert(
   System.Text.Encoding srcEncoding,
   System.Text.Encoding dstEncoding,
   ubyte[] bytes
);
public static ubyte[] Convert(
   System.Text.Encoding srcEncoding,
   System.Text.Encoding dstEncoding,
   ubyte[] bytes,
   int index,
   int count
);
public boolean Equals(
   System.Object value
);
public int GetByteCount(
   char[] chars
);
public int GetByteCount(
   System.String s
);
public abstract int GetByteCount(
   char[] chars,
   int index,
   int count
);
public ubyte[] GetBytes(
   char[] chars
);
public ubyte[] GetBytes(
```

```
   System.String s
);
public ubyte[] GetBytes(
   char[] chars,
   int index,
   int count
);
public abstract int GetBytes(
   char[] chars,
   int charIndex,
   int charCount,
   ubyte[] bytes,
   int byteIndex
);
public int GetBytes(
   System.String s,
   int charIndex,
   int charCount,
   ubyte[] bytes,
   int byteIndex
);
public int GetCharCount(
   ubyte[] bytes
);
public abstract int GetCharCount(
   ubyte[] bytes,
   int index,
   int count
);
public char[] GetChars(
   ubyte[] bytes
);
public char[] GetChars(
   ubyte[] bytes,
   int index,
   int count
);
public abstract int GetChars(
   ubyte[] bytes,
   int byteIndex,
   int byteCount,
   char[] chars,
   int charIndex
);
public System.Text.Decoder GetDecoder();
public System.Text.Encoder GetEncoder();
public static System.Text.Encoding GetEncoding(
   int codepage
);
public static System.Text.Encoding GetEncoding(
   System.String name
);
public int GetHashCode();
public abstract int GetMaxByteCount(
   int charCount
);
public abstract int GetMaxCharCount(
   int byteCount
);
public ubyte[] GetPreamble();
public System.String GetString(
   ubyte[] bytes
);
public System.String GetString(
   ubyte[] bytes,
   int index,
   int count
);
```

Protected Constructors

```
protected Encoding();
protected Encoding(
   int codePage
);
```

StringBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Chars") */
public class StringBuilder
   extends System.Object
```

Public Constructors

```
public StringBuilder();
public StringBuilder(
   int capacity
);
public StringBuilder(
   System.String value
);
public StringBuilder(
   int capacity,
   int maxCapacity
);
public StringBuilder(
   System.String value,
   int capacity
);
public StringBuilder(
   System.String value,
   int startIndex,
   int length,
   int capacity
);
```

Public Properties

```
public int get_Capacity();
public void set_Capacity(
   int value
);
public char get_Chars(
   int index
);
public void set_Chars(
   int index,
   char value
);
public int get_Length();
public void set_Length(
   int value
);
public int get_MaxCapacity();
```

Public Methods

```
public System.Text.StringBuilder Append(
   boolean value
);
```

```
public System.Text.StringBuilder Append(
   ubyte value
);
public System.Text.StringBuilder Append(
   char value
);
public System.Text.StringBuilder Append(
   char[] value
);
public System.Text.StringBuilder Append(
   double value
);
public System.Text.StringBuilder Append(
   short value
);
public System.Text.StringBuilder Append(
   int value
);
public System.Text.StringBuilder Append(
   long value
);
public System.Text.StringBuilder Append(
   float value
);
public System.Text.StringBuilder Append(
   System.Decimal value
);
public System.Text.StringBuilder Append(
   System.Object value
);
public System.Text.StringBuilder Append(
   System.String value
);
public System.Text.StringBuilder Append(
   char value,
   int repeatCount
);
public System.Text.StringBuilder Append(
   char[] value,
   int startIndex,
   int charCount
);
public System.Text.StringBuilder Append(
   System.String value,
   int startIndex,
   int count
);
public System.Text.StringBuilder AppendFormat(
   System.String format,
   System.Object arg0
);
public System.Text.StringBuilder AppendFormat(
   System.String format,
   System.Object[] args
);
public System.Text.StringBuilder AppendFormat(
   System.IFormatProvider provider,
   System.String format,
   System.Object[] args
);
public System.Text.StringBuilder AppendFormat(
   System.String format,
   System.Object arg0,
   System.Object arg1
);
public System.Text.StringBuilder AppendFormat(
   System.String format,
   System.Object arg0,
```

```
   System.Object arg1,
   System.Object arg2
);
public int EnsureCapacity(
   int capacity
);
public boolean Equals(
   System.Text.StringBuilder sb
);
public System.Text.StringBuilder Insert(
   int index,
   boolean value
);
public System.Text.StringBuilder Insert(
   int index,
   ubyte value
);
public System.Text.StringBuilder Insert(
   int index,
   char value
);
public System.Text.StringBuilder Insert(
   int index,
   char[] value
);
public System.Text.StringBuilder Insert(
   int index,
   double value
);
public System.Text.StringBuilder Insert(
   int index,
   short value
);
public System.Text.StringBuilder Insert(
   int index,
   int value
);
public System.Text.StringBuilder Insert(
   int index,
   long value
);
public System.Text.StringBuilder Insert(
   int index,
   float value
);
public System.Text.StringBuilder Insert(
   int index,
   System.Decimal value
);
public System.Text.StringBuilder Insert(
   int index,
   System.Object value
);
public System.Text.StringBuilder Insert(
   int index,
   System.String value
);
public System.Text.StringBuilder Insert(
   int index,
   System.String value,
   int count
);
public System.Text.StringBuilder Insert(
   int index,
   char[] value,
   int startIndex,
   int charCount
);
```

```
public System.Text.StringBuilder Remove(
   int startIndex,
   int length
);
public System.Text.StringBuilder Replace(
   char oldChar,
   char newChar
);
public System.Text.StringBuilder Replace(
   System.String oldValue,
   System.String newValue
);
public System.Text.StringBuilder Replace(
   char oldChar,
   char newChar,
   int startIndex,
   int count
);
public System.Text.StringBuilder Replace(
   System.String oldValue,
   System.String newValue,
   int startIndex,
   int count
);
public System.String ToString();
public System.String ToString(
   int startIndex,
   int length
);
```

UnicodeEncoding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class UnicodeEncoding
   extends System.Text.Encoding
```

Public Constructors

```
public UnicodeEncoding();
public UnicodeEncoding(
   boolean bigEndian,
   boolean byteOrderMark
);
```

Public Methods

```
public boolean Equals(
   System.Object value
);
public int GetByteCount(
   System.String s
);
public int GetByteCount(
   char[] chars,
   int index,
   int count
);
public ubyte[] GetBytes(
   System.String s
);
public int GetBytes(
   char[] chars,
   int charIndex,
   int charCount,
   ubyte[] bytes,
   int byteIndex
);
public int GetBytes(
   System.String s,
   int charIndex,
   int charCount,
   ubyte[] bytes,
   int byteIndex
);
public int GetCharCount(
   ubyte[] bytes,
   int index,
   int count
);
public int GetChars(
   ubyte[] bytes,
   int byteIndex,
   int byteCount,
   char[] chars,
   int charIndex
);
public System.Text.Decoder GetDecoder();
```

```
public int GetHashCode();
public int GetMaxByteCount(
   int charCount
);
public int GetMaxCharCount(
   int byteCount
);
public ubyte[] GetPreamble();
```

Public Fields

```
public static int CharSize;
```

UTF7Encoding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class UTF7Encoding
   extends System.Text.Encoding
```

Public Constructors

```
public UTF7Encoding();
public UTF7Encoding(
   boolean allowOptionals
);
```

Public Methods

```
public int GetByteCount(
   char[] chars,
   int index,
   int count
);
public int GetBytes(
   char[] chars,
   int charIndex,
   int charCount,
   ubyte[] bytes,
   int byteIndex
);
public int GetCharCount(
   ubyte[] bytes,
   int index,
   int count
);
public int GetChars(
   ubyte[] bytes,
   int byteIndex,
   int byteCount,
   char[] chars,
   int charIndex
);
public System.Text.Decoder GetDecoder();
public System.Text.Encoder GetEncoder();
public int GetMaxByteCount(
   int charCount
);
public int GetMaxCharCount(
   int byteCount
);
```

UTF8Encoding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class UTF8Encoding
   extends System.Text.Encoding
```

Public Constructors

```
public UTF8Encoding();
public UTF8Encoding(
   boolean encoderShouldEmitUTF8Identifier
);
public UTF8Encoding(
   boolean encoderShouldEmitUTF8Identifier,
   boolean throwOnInvalidBytes
);
```

Public Methods

```
public boolean Equals(
   System.Object value
);
public int GetByteCount(
   System.String chars
);
public int GetByteCount(
   char[] chars,
   int index,
   int count
);
public ubyte[] GetBytes(
   System.String s
);
public int GetBytes(
   char[] chars,
   int charIndex,
   int charCount,
   ubyte[] bytes,
   int byteIndex
);
public int GetBytes(
   System.String s,
   int charIndex,
   int charCount,
   ubyte[] bytes,
   int byteIndex
);
public int GetCharCount(
   ubyte[] bytes,
   int index,
   int count
);
public int GetChars(
   ubyte[] bytes,
   int byteIndex,
   int byteCount,
   char[] chars,
```

```
      int charIndex
   );
   public System.Text.Decoder GetDecoder();
   public System.Text.Encoder GetEncoder();
   public int GetHashCode();
   public int GetMaxByteCount(
      int charCount
   );
   public int GetMaxCharCount(
      int byteCount
   );
   public ubyte[] GetPreamble();
```

System.Text.RegularExpressions Namespace (Visual J# Syntax)

Classes

Capture

CaptureCollection

Group

GroupCollection

Match

MatchCollection

Regex

RegexCompilationInfo

RegexRunner

RegexRunnerFactory

Delegates

MatchEvaluator

Enumerations

RegexOptions

Capture Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Capture
   extends System.Object
```

Public Properties

```
public int get_Index();
public int get_Length();
public System.String get_Value();
```

Public Methods

```
public System.String ToString();
```

CaptureCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CaptureCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Text.RegularExpressions.Capture get_Item(
   int i
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public final void CopyTo(
   System.Array array,
   int arrayIndex
);
public final System.Collections.IEnumerator GetEnumerator();
```

Group Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Group
   extends System.Text.RegularExpressions.Capture
```

Public Properties

```
public System.Text.RegularExpressions.CaptureCollection get_Captures();
public boolean get_Success();
```

Public Methods

```
public static System.Text.RegularExpressions.Group Synchronized(
   System.Text.RegularExpressions.Group inner
);
```

GroupCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class GroupCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Text.RegularExpressions.Group get_Item(
   int groupnum
);
public System.Text.RegularExpressions.Group get_Item(
   System.String groupname
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public final void CopyTo(
   System.Array array,
   int arrayIndex
);
public final System.Collections.IEnumerator GetEnumerator();
```

Match Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Match
   extends System.Text.RegularExpressions.Group
```

Public Properties

```
public static System.Text.RegularExpressions.Match get_Empty();
public System.Text.RegularExpressions.GroupCollection get_Groups();
```

Public Methods

```
public System.Text.RegularExpressions.Match NextMatch();
public System.String Result(
   System.String replacement
);
public static System.Text.RegularExpressions.Match Synchronized(
   System.Text.RegularExpressions.Match inner
);
```

MatchCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class MatchCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Text.RegularExpressions.Match get_Item(
   int i
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public final void CopyTo(
   System.Array array,
   int arrayIndex
);
public final System.Collections.IEnumerator GetEnumerator();
```

MatchEvaluator Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
System.String MatchEvaluator(
   System.Text.RegularExpressions.Match match
);
```

Regex Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class Regex
   extends System.Object
   implements System.Runtime.Serialization.ISerializable
```

Public Constructors

```
public Regex(
   System.String pattern
);
public Regex(
   System.String pattern,
   System.Text.RegularExpressions.RegexOptions options
);
```

Public Properties

```
public System.Text.RegularExpressions.RegexOptions get_Options();
public boolean get_RightToLeft();
```

Public Methods

```
public static void CompileToAssembly(
   System.Text.RegularExpressions.RegexCompilationInfo[] regexinfos,
   System.Reflection.AssemblyName assemblyname
);
public static void CompileToAssembly(
   System.Text.RegularExpressions.RegexCompilationInfo[] regexinfos,
   System.Reflection.AssemblyName assemblyname,
   System.Reflection.Emit.CustomAttributeBuilder[] attributes
);
public static void CompileToAssembly(
   System.Text.RegularExpressions.RegexCompilationInfo[] regexinfos,
   System.Reflection.AssemblyName assemblyname,
   System.Reflection.Emit.CustomAttributeBuilder[] attributes,
   System.String resourceFile
);
public static System.String Escape(
   System.String str
);
public System.String[] GetGroupNames();
public int[] GetGroupNumbers();
public System.String GroupNameFromNumber(
   int i
);
public int GroupNumberFromName(
   System.String name
);
public boolean IsMatch(
   System.String input
);
public boolean IsMatch(
   System.String input,
```

```
   int startat
);
public static boolean IsMatch(
   System.String input,
   System.String pattern
);
public static boolean IsMatch(
   System.String input,
   System.String pattern,
   System.Text.RegularExpressions.RegexOptions options
);
public System.Text.RegularExpressions.Match Match(
   System.String input
);
public System.Text.RegularExpressions.Match Match(
   System.String input,
   int startat
);
public static System.Text.RegularExpressions.Match Match(
   System.String input,
   System.String pattern
);
public System.Text.RegularExpressions.Match Match(
   System.String input,
   int beginning,
   int length
);
public static System.Text.RegularExpressions.Match Match(
   System.String input,
   System.String pattern,
   System.Text.RegularExpressions.RegexOptions options
);
public System.Text.RegularExpressions.MatchCollection Matches(
   System.String input
);
public System.Text.RegularExpressions.MatchCollection Matches(
   System.String input,
   int startat
);
public static System.Text.RegularExpressions.MatchCollection Matches(
   System.String input,
   System.String pattern
);
public static System.Text.RegularExpressions.MatchCollection Matches(
   System.String input,
   System.String pattern,
   System.Text.RegularExpressions.RegexOptions options
);
public System.String Replace(
   System.String input,
   System.String replacement
);
public System.String Replace(
   System.String input,
   System.Text.RegularExpressions.MatchEvaluator evaluator
);
public System.String Replace(
   System.String input,
   System.String replacement,
   int count
);
public static System.String Replace(
   System.String input,
   System.String pattern,
   System.String replacement
);
public static System.String Replace(
   System.String input,
```

```
   System.String pattern,
   System.Text.RegularExpressions.MatchEvaluator evaluator
);
public System.String Replace(
   System.String input,
   System.Text.RegularExpressions.MatchEvaluator evaluator,
   int count
);
public System.String Replace(
   System.String input,
   System.String replacement,
   int count,
   int startat
);
public static System.String Replace(
   System.String input,
   System.String pattern,
   System.String replacement,
   System.Text.RegularExpressions.RegexOptions options
);
public static System.String Replace(
   System.String input,
   System.String pattern,
   System.Text.RegularExpressions.MatchEvaluator evaluator,
   System.Text.RegularExpressions.RegexOptions options
);
public System.String Replace(
   System.String input,
   System.Text.RegularExpressions.MatchEvaluator evaluator,
   int count,
   int startat
);
public System.String[] Split(
   System.String input
);
public System.String[] Split(
   System.String input,
   int count
);
public static System.String[] Split(
   System.String input,
   System.String pattern
);
public System.String[] Split(
   System.String input,
   int count,
   int startat
);
public static System.String[] Split(
   System.String input,
   System.String pattern,
   System.Text.RegularExpressions.RegexOptions options
);
public System.String ToString();
public static System.String Unescape(
   System.String str
);
```

Protected Constructors

```
protected Regex();
```

Protected Methods

```
protected void Finalize();
protected void InitializeReferences();
protected boolean UseOptionC();
protected boolean UseOptionR();
```

Protected Fields

```
protected System.Collections.Hashtable capnames;
protected System.Collections.Hashtable caps;
protected int capsize;
protected System.String[] capslist;
protected System.Text.RegularExpressions.RegexRunnerFactory factory;
protected System.String pattern;
protected System.Text.RegularExpressions.RegexOptions roptions;
```

RegexCompilationInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class RegexCompilationInfo
   extends System.Object
```

Public Constructors

```
public RegexCompilationInfo(
   System.String pattern,
   System.Text.RegularExpressions.RegexOptions options,
   System.String name,
   System.String fullnamespace,
   boolean ispublic
);
```

Public Properties

```
public boolean get_IsPublic();
public void set_IsPublic(
   boolean value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.Text.RegularExpressions.RegexOptions get_Options();
public void set_Options(
   System.Text.RegularExpressions.RegexOptions value
);
public System.String get_Pattern();
public void set_Pattern(
   System.String value
);
```

RegexOptions Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum RegexOptions
```

Members

Compiled

ECMAScript

ExplicitCapture

IgnoreCase

IgnorePatternWhitespace

Multiline

None

RightToLeft

Singleline

RegexRunner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public abstract class RegexRunner
   extends System.Object
```

Protected Constructors

```
protected RegexRunner();
```

Protected Methods

```
protected void Capture(
   int capnum,
   int start,
   int end
);
protected static boolean CharInSet(
   char ch,
   System.String set,
   System.String category
);
protected void Crawl(
   int i
);
protected int Crawlpos();
protected void DoubleCrawl();
protected void DoubleStack();
protected void DoubleTrack();
protected void EnsureStorage();
protected abstract boolean FindFirstChar();
protected abstract void Go();
protected abstract void InitTrackCount();
protected boolean IsBoundary(
   int index,
   int startpos,
   int endpos
);
protected boolean IsECMABoundary(
   int index,
   int startpos,
   int endpos
);
protected boolean IsMatched(
   int cap
);
protected int MatchIndex(
   int cap
);
protected int MatchLength(
   int cap
);
protected int Popcrawl();
protected System.Text.RegularExpressions.Match Scan(
   System.Text.RegularExpressions.Regex regex,
   System.String text,
   int textbeg,
```

```
    int textend,
    int textstart,
    int prevlen,
    boolean quick
);
protected void TransferCapture(
    int capnum,
    int uncapnum,
    int start,
    int end
);
protected void Uncapture();
```

Protected Fields

```
protected int[] runcrawl;
protected int runcrawlpos;
protected System.Text.RegularExpressions.Match runmatch;
protected System.Text.RegularExpressions.Regex runregex;
protected int[] runstack;
protected int runstackpos;
protected System.String runtext;
protected int runtextbeg;
protected int runtextend;
protected int runtextpos;
protected int runtextstart;
protected int[] runtrack;
protected int runtrackcount;
protected int runtrackpos;
```

RegexRunnerFactory Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public abstract class RegexRunnerFactory
   extends System.Object
```

Protected Constructors

```
protected RegexRunnerFactory();
```

Protected Methods

```
protected abstract System.Text.RegularExpressions.RegexRunner CreateInstance();
```

System.Threading Namespace (Visual J# Syntax)

Classes

AutoResetEvent

CompressedStack

Interlocked

ManualResetEvent

Monitor

Mutex

Overlapped

ReaderWriterLock

RegisteredWaitHandle

SynchronizationLockException

Thread

ThreadAbortException

ThreadExceptionEventArgs

ThreadInterruptedException

ThreadPool

ThreadStateException

Timeout

Timer

WaitHandle

Structures

LockCookie

NativeOverlapped

Delegates

IOCompletionCallback

ThreadExceptionEventHandler

ThreadStart

TimerCallback

WaitCallback

WaitOrTimerCallback

Enumerations

ApartmentState

ThreadPriority

ThreadState

ApartmentState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ApartmentState
```

Members

MTA

STA

Unknown

AutoResetEvent Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AutoResetEvent
   extends System.Threading.WaitHandle
```

Public Constructors

```
public AutoResetEvent(
   boolean initialState
);
```

Public Methods

```
public boolean Reset();
public boolean Set();
```

CompressedStack Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CompressedStack
   extends System.Object
```

Protected Methods

```
protected void Finalize();
```

Interlocked Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Interlocked
   extends System.Object
```

Public Methods

```
public static int CompareExchange(
   int location1,
   int value,
   int comparand
);
public static float CompareExchange(
   float location1,
   float value,
   float comparand
);
public static System.Object CompareExchange(
   System.Object& location1,
   System.Object value,
   System.Object comparand
);
public static int Decrement(
   int location
);
public static long Decrement(
   long location
);
public static int Exchange(
   int location1,
   int value
);
public static float Exchange(
   float location1,
   float value
);
public static System.Object Exchange(
   System.Object& location1,
   System.Object value
);
public static int Increment(
   int location
);
public static long Increment(
   long location
);
```

IOCompletionCallback Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void IOCompletionCallback(
   System.UInt32 errorCode,
   System.UInt32 numBytes,
   System.Threading.NativeOverlapped* pOVERLAP
);
```

LockCookie Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class LockCookie
   extends System.ValueType
```

ManualResetEvent Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ManualResetEvent
   extends System.Threading.WaitHandle
```

Public Constructors

```
public ManualResetEvent(
   boolean initialState
);
```

Public Methods

```
public boolean Reset();
public boolean Set();
```

Monitor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Monitor
   extends System.Object
```

Public Methods

```
public static void Enter(
   System.Object obj
);
public static void Exit(
   System.Object obj
);
public static void Pulse(
   System.Object obj
);
public static void PulseAll(
   System.Object obj
);
public static boolean TryEnter(
   System.Object obj
);
public static boolean TryEnter(
   System.Object obj,
   int millisecondsTimeout
);
public static boolean TryEnter(
   System.Object obj,
   System.TimeSpan timeout
);
public static boolean Wait(
   System.Object obj
);
public static boolean Wait(
   System.Object obj,
   int millisecondsTimeout
);
public static boolean Wait(
   System.Object obj,
   System.TimeSpan timeout
);
public static boolean Wait(
   System.Object obj,
   int millisecondsTimeout,
   boolean exitContext
);
public static boolean Wait(
   System.Object obj,
   System.TimeSpan timeout,
   boolean exitContext
);
```

Mutex Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Mutex
   extends System.Threading.WaitHandle
```

Public Constructors

```
public Mutex();
public Mutex(
   boolean initiallyOwned
);
public Mutex(
   boolean initiallyOwned,
   System.String name
);
public Mutex(
   boolean initiallyOwned,
   System.String name,
   boolean createdNew
);
```

Public Methods

```
public void ReleaseMutex();
```

NativeOverlapped Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class NativeOverlapped
   extends System.ValueType
```

Public Fields

```
public int EventHandle;
public int InternalHigh;
public int InternalLow;
public int OffsetHigh;
public int OffsetLow;
```

Overlapped Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Overlapped
   extends System.Object
```

Public Constructors

```
public Overlapped();
public Overlapped(
   int offsetLo,
   int offsetHi,
   int hEvent,
   System.IAsyncResult ar
);
```

Public Properties

```
public System.IAsyncResult get_AsyncResult();
public void set_AsyncResult(
   System.IAsyncResult value
);
public int get_EventHandle();
public void set_EventHandle(
   int value
);
public int get_OffsetHigh();
public void set_OffsetHigh(
   int value
);
public int get_OffsetLow();
public void set_OffsetLow(
   int value
);
```

ReaderWriterLock Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ReaderWriterLock
   extends System.Object
```

Public Constructors

```
public ReaderWriterLock();
```

Public Properties

```
public boolean get_IsReaderLockHeld();
public boolean get_IsWriterLockHeld();
public int get_WriterSeqNum();
```

Public Methods

```
public void AcquireReaderLock(
   int millisecondsTimeout
);
public void AcquireReaderLock(
   System.TimeSpan timeout
);
public void AcquireWriterLock(
   int millisecondsTimeout
);
public void AcquireWriterLock(
   System.TimeSpan timeout
);
public boolean AnyWritersSince(
   int seqNum
);
public void DowngradeFromWriterLock(
   System.Threading.LockCookie& lockCookie
);
public System.Threading.LockCookie ReleaseLock();
public void ReleaseReaderLock();
public void ReleaseWriterLock();
public void RestoreLock(
   System.Threading.LockCookie& lockCookie
);
public System.Threading.LockCookie UpgradeToWriterLock(
   int millisecondsTimeout
);
public System.Threading.LockCookie UpgradeToWriterLock(
   System.TimeSpan timeout
);
```

RegisteredWaitHandle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RegisteredWaitHandle
   extends System.MarshalByRefObject
```

Public Methods

```
public boolean Unregister(
   System.Threading.WaitHandle waitObject
);
```

Protected Methods

```
protected void Finalize();
```

SynchronizationLockException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class SynchronizationLockException
   extends System.SystemException
```

Public Constructors

```
public SynchronizationLockException();
public SynchronizationLockException(
   System.String message
);
public SynchronizationLockException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected SynchronizationLockException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Thread Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Thread
   extends System.Object
```

Public Constructors

```
public Thread(
   System.Threading.ThreadStart start
);
```

Public Properties

```
public System.Threading.ApartmentState get_ApartmentState();
public void set_ApartmentState(
   System.Threading.ApartmentState value
);
public static System.Runtime.Remoting.Contexts.Context get_CurrentContext();
public System.Globalization.CultureInfo get_CurrentCulture();
public void set_CurrentCulture(
   System.Globalization.CultureInfo value
);
public static System.Security.Principal.IPrincipal get_CurrentPrincipal();
public static void set_CurrentPrincipal(
   System.Security.Principal.IPrincipal value
);
public static System.Threading.Thread get_CurrentThread();
public System.Globalization.CultureInfo get_CurrentUICulture();
public void set_CurrentUICulture(
   System.Globalization.CultureInfo value
);
public boolean get_IsAlive();
public boolean get_IsBackground();
public void set_IsBackground(
   boolean value
);
public boolean get_IsThreadPoolThread();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Threading.ThreadPriority get_Priority();
public void set_Priority(
   System.Threading.ThreadPriority value
);
public System.Threading.ThreadState get_ThreadState();
```

Public Methods

```
public void Abort();
public void Abort(
   System.Object stateInfo
);
public static System.LocalDataStoreSlot AllocateDataSlot();
public static System.LocalDataStoreSlot AllocateNamedDataSlot(
```

```
      System.String name
   );
   public static void FreeNamedDataSlot(
      System.String name
   );
   public static System.Object GetData(
      System.LocalDataStoreSlot slot
   );
   public static System.AppDomain GetDomain();
   public static int GetDomainID();
   public static System.LocalDataStoreSlot GetNamedDataSlot(
      System.String name
   );
   public void Interrupt();
   public void Join();
   public boolean Join(
      int millisecondsTimeout
   );
   public boolean Join(
      System.TimeSpan timeout
   );
   public static void ResetAbort();
   public void Resume();
   public static void SetData(
      System.LocalDataStoreSlot slot,
      System.Object data
   );
   public static void Sleep(
      int millisecondsTimeout
   );
   public static void Sleep(
      System.TimeSpan timeout
   );
   public static void SpinWait(
      int iterations
   );
   public void Start();
   public void Suspend();
```

Protected Methods

```
   protected void Finalize();
```

ThreadAbortException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ThreadAbortException
   extends System.SystemException
```

Public Properties

```
public System.Object get_ExceptionState();
```

ThreadExceptionEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ThreadExceptionEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ThreadExceptionEventArgs(
   System.Exception t
);
```

Public Properties

```
public System.Exception get_Exception();
```

ThreadExceptionEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ThreadExceptionEventHandler(
   System.Object sender,
   System.Threading.ThreadExceptionEventArgs e
);
```

ThreadInterruptedException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ThreadInterruptedException
   extends System.SystemException
```

Public Constructors

```
public ThreadInterruptedException();
public ThreadInterruptedException(
   System.String message
);
public ThreadInterruptedException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected ThreadInterruptedException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ThreadPool Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ThreadPool
   extends System.Object
```

Public Methods

```
public static boolean BindHandle(
   System.IntPtr osHandle
);
public static void GetAvailableThreads(
   int workerThreads,
   int completionPortThreads
);
public static void GetMaxThreads(
   int workerThreads,
   int completionPortThreads
);
public static boolean QueueUserWorkItem(
   System.Threading.WaitCallback callBack
);
public static boolean QueueUserWorkItem(
   System.Threading.WaitCallback callBack,
   System.Object state
);
public static System.Threading.RegisteredWaitHandle RegisterWaitForSingleObject(
   System.Threading.WaitHandle waitObject,
   System.Threading.WaitOrTimerCallback callBack,
   System.Object state,
   int millisecondsTimeOutInterval,
   boolean executeOnlyOnce
);
public static System.Threading.RegisteredWaitHandle RegisterWaitForSingleObject(
   System.Threading.WaitHandle waitObject,
   System.Threading.WaitOrTimerCallback callBack,
   System.Object state,
   long millisecondsTimeOutInterval,
   boolean executeOnlyOnce
);
public static System.Threading.RegisteredWaitHandle RegisterWaitForSingleObject(
   System.Threading.WaitHandle waitObject,
   System.Threading.WaitOrTimerCallback callBack,
   System.Object state,
   System.TimeSpan timeout,
   boolean executeOnlyOnce
);
public static boolean UnsafeQueueUserWorkItem(
   System.Threading.WaitCallback callBack,
   System.Object state
);
public static System.Threading.RegisteredWaitHandle UnsafeRegisterWaitForSingleObject(
   System.Threading.WaitHandle waitObject,
   System.Threading.WaitOrTimerCallback callBack,
   System.Object state,
   int millisecondsTimeOutInterval,
   boolean executeOnlyOnce
);
public static System.Threading.RegisteredWaitHandle UnsafeRegisterWaitForSingleObject(
   System.Threading.WaitHandle waitObject,
```

```
    System.Threading.WaitOrTimerCallback callBack,
    System.Object state,
    long millisecondsTimeOutInterval,
    boolean executeOnlyOnce
);
public static System.Threading.RegisteredWaitHandle UnsafeRegisterWaitForSingleObject(
    System.Threading.WaitHandle waitObject,
    System.Threading.WaitOrTimerCallback callBack,
    System.Object state,
    System.TimeSpan timeout,
    boolean executeOnlyOnce
);
```

ThreadPriority Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ThreadPriority
```

Members

AboveNormal

BelowNormal

Highest

Lowest

Normal

ThreadStart Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ThreadStart();
```

ThreadState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum ThreadState
```

Members

Aborted

AbortRequested

Background

Running

Stopped

StopRequested

Suspended

SuspendRequested

Unstarted

WaitSleepJoin

ThreadStateException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ThreadStateException
   extends System.SystemException
```

Public Constructors

```
public ThreadStateException();
public ThreadStateException(
   System.String message
);
public ThreadStateException(
   System.String message,
   System.Exception innerException
);
```

Protected Constructors

```
protected ThreadStateException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Timeout Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Timeout
   extends System.Object
```

Public Fields

```
public static int Infinite;
```

Timer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Timer
   extends System.MarshalByRefObject
   implements System.IDisposable
```

Public Constructors

```
public Timer(
   System.Threading.TimerCallback callback,
   System.Object state,
   int dueTime,
   int period
);
public Timer(
   System.Threading.TimerCallback callback,
   System.Object state,
   long dueTime,
   long period
);
public Timer(
   System.Threading.TimerCallback callback,
   System.Object state,
   System.TimeSpan dueTime,
   System.TimeSpan period
);
```

Public Methods

```
public boolean Change(
   int dueTime,
   int period
);
public boolean Change(
   long dueTime,
   long period
);
public boolean Change(
   System.TimeSpan dueTime,
   System.TimeSpan period
);
public final void Dispose();
public boolean Dispose(
   System.Threading.WaitHandle notifyObject
);
```

Protected Methods

```
protected void Finalize();
```

TimerCallback Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void TimerCallback(
   System.Object state
);
```

WaitCallback Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void WaitCallback(
   System.Object state
);
```

WaitHandle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class WaitHandle
   extends System.MarshalByRefObject
   implements System.IDisposable
```

Public Constructors

```
public WaitHandle();
```

Public Properties

```
public System.IntPtr get_Handle();
public void set_Handle(
   System.IntPtr value
);
```

Public Methods

```
public void Close();
public static boolean WaitAll(
   System.Threading.WaitHandle[] waitHandles
);
public static boolean WaitAll(
   System.Threading.WaitHandle[] waitHandles,
   int millisecondsTimeout,
   boolean exitContext
);
public static boolean WaitAll(
   System.Threading.WaitHandle[] waitHandles,
   System.TimeSpan timeout,
   boolean exitContext
);
public static int WaitAny(
   System.Threading.WaitHandle[] waitHandles
);
public static int WaitAny(
   System.Threading.WaitHandle[] waitHandles,
   int millisecondsTimeout,
   boolean exitContext
);
public static int WaitAny(
   System.Threading.WaitHandle[] waitHandles,
   System.TimeSpan timeout,
   boolean exitContext
);
public boolean WaitOne();
public boolean WaitOne(
   int millisecondsTimeout,
   boolean exitContext
);
public boolean WaitOne(
   System.TimeSpan timeout,
   boolean exitContext
);
```

Public Fields

```
public static int WaitTimeout;
```

Protected Methods

```
protected void Dispose(
   boolean explicitDisposing
);
protected void Finalize();
```

Protected Fields

```
protected static final System.IntPtr InvalidHandle;
```

WaitOrTimerCallback Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void WaitOrTimerCallback(
   System.Object state,
   boolean timedOut
);
```

System.Timers Namespace (Visual J# Syntax)

Classes

ElapsedEventArgs

Timer

TimersDescriptionAttribute

Delegates

ElapsedEventHandler

ElapsedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ElapsedEventArgs
   extends System.EventArgs
```

Public Properties

```
public System.DateTime get_SignalTime();
```

ElapsedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ElapsedEventHandler(
   System.Object sender,
   System.Timers.ElapsedEventArgs e
);
```

Timer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Interval") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Elapsed") */
public class Timer
   extends System.ComponentModel.Component
   implements System.ComponentModel.ISupportInitialize
```

Public Constructors

```
public Timer();
public Timer(
   double interval
);
```

Public Properties

```
public boolean get_AutoReset();
public void set_AutoReset(
   boolean value
);
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public double get_Interval();
public void set_Interval(
   double value
);
public System.ComponentModel.ISite get_Site();
public void set_Site(
   System.ComponentModel.ISite value
);
public System.ComponentModel.ISynchronizeInvoke get_SynchronizingObject();
public void set_SynchronizingObject(
   System.ComponentModel.ISynchronizeInvoke value
);
```

Public Methods

```
public final void BeginInit();
public void Close();
public final void EndInit();
public void Start();
public void Stop();
```

Public Events

```
public void add_Elapsed(
   System.Timers.ElapsedEventHandler value
);
public void remove_Elapsed(
   System.Timers.ElapsedEventHandler value
```

```
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

TimersDescriptionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class TimersDescriptionAttribute
   extends System.ComponentModel.DescriptionAttribute
```

Public Constructors

```
public TimersDescriptionAttribute(
   System.String description
);
```

Public Properties

```
public System.String get_Description();
```

System.Web Namespace (Visual J# Syntax)

Classes

HttpApplication

HttpApplicationState

HttpBrowserCapabilities

HttpCachePolicy

HttpCacheVaryByHeaders

HttpCacheVaryByParams

HttpClientCertificate

HttpCompileException

HttpContext

HttpCookie

HttpCookieCollection

HttpException

HttpFileCollection

HttpModuleCollection

HttpParseException

HttpPostedFile

HttpRequest

HttpResponse

HttpRuntime

HttpServerUtility

HttpStaticObjectsCollection

HttpUnhandledException

HttpUtility

HttpWorkerRequest

HttpWriter

ProcessInfo

ProcessModelInfo

TraceContext

Interfaces

IHttpAsyncHandler

IHttpHandler

IHttpHandlerFactory

IHttpModule

Delegates

BeginEventHandler

EndEventHandler

HttpCacheValidateHandler

HttpWorkerRequest.EndOfSendNotification

Enumerations

HttpCacheability

HttpCacheRevalidation

HttpValidationStatus

ProcessShutdownReason

ProcessStatus

TraceMode

BeginEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
System.IAsyncResult BeginEventHandler(
   System.Object sender,
   System.EventArgs e,
   System.AsyncCallback cb,
   System.Object extraData
);
```

EndEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void EndEventHandler(
   System.IAsyncResult ar
);
```

HttpApplication Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class HttpApplication
   extends System.Object
   implements System.Web.IHttpAsyncHandler, System.Web.IHttpHandler, System.ComponentModel.IC
omponent, System.IDisposable
```

Public Constructors

```
public HttpApplication();
```

Public Properties

```
public System.Web.HttpApplicationState get_Application();
public System.Web.HttpContext get_Context();
public System.Web.HttpModuleCollection get_Modules();
public System.Web.HttpRequest get_Request();
public System.Web.HttpResponse get_Response();
public System.Web.HttpServerUtility get_Server();
public System.Web.SessionState.HttpSessionState get_Session();
public final System.ComponentModel.ISite get_Site();
public final void set_Site(
   System.ComponentModel.ISite value
);
public System.Security.Principal.IPrincipal get_User();
```

Public Methods

```
public void AddOnAcquireRequestStateAsync(
   System.Web.BeginEventHandler bh,
   System.Web.EndEventHandler eh
);
public void AddOnAuthenticateRequestAsync(
   System.Web.BeginEventHandler bh,
   System.Web.EndEventHandler eh
);
public void AddOnAuthorizeRequestAsync(
   System.Web.BeginEventHandler bh,
   System.Web.EndEventHandler eh
);
public void AddOnBeginRequestAsync(
   System.Web.BeginEventHandler bh,
   System.Web.EndEventHandler eh
);
public void AddOnEndRequestAsync(
   System.Web.BeginEventHandler bh,
   System.Web.EndEventHandler eh
);
public void AddOnPostRequestHandlerExecuteAsync(
   System.Web.BeginEventHandler bh,
   System.Web.EndEventHandler eh
);
public void AddOnPreRequestHandlerExecuteAsync(
   System.Web.BeginEventHandler bh,
```

```
   System.Web.EndEventHandler eh
);
public void AddOnReleaseRequestStateAsync(
   System.Web.BeginEventHandler bh,
   System.Web.EndEventHandler eh
);
public void AddOnResolveRequestCacheAsync(
   System.Web.BeginEventHandler bh,
   System.Web.EndEventHandler eh
);
public void AddOnUpdateRequestCacheAsync(
   System.Web.BeginEventHandler bh,
   System.Web.EndEventHandler eh
);
public void CompleteRequest();
public void Dispose();
public System.String GetVaryByCustomString(
   System.Web.HttpContext context,
   System.String custom
);
public void Init();
```

Public Events

```
public void add_AcquireRequestState(
   System.EventHandler value
);
public void remove_AcquireRequestState(
   System.EventHandler value
);
public void add_AuthenticateRequest(
   System.EventHandler value
);
public void remove_AuthenticateRequest(
   System.EventHandler value
);
public void add_AuthorizeRequest(
   System.EventHandler value
);
public void remove_AuthorizeRequest(
   System.EventHandler value
);
public void add_BeginRequest(
   System.EventHandler value
);
public void remove_BeginRequest(
   System.EventHandler value
);
public final void add_Disposed(
   System.EventHandler value
);
public final void remove_Disposed(
   System.EventHandler value
);
public void add_EndRequest(
   System.EventHandler value
);
public void remove_EndRequest(
   System.EventHandler value
);
public void add_Error(
   System.EventHandler value
);
public void remove_Error(
   System.EventHandler value
);
```

```
public void add_PostRequestHandlerExecute(
   System.EventHandler value
);
public void remove_PostRequestHandlerExecute(
   System.EventHandler value
);
public void add_PreRequestHandlerExecute(
   System.EventHandler value
);
public void remove_PreRequestHandlerExecute(
   System.EventHandler value
);
public void add_PreSendRequestContent(
   System.EventHandler value
);
public void remove_PreSendRequestContent(
   System.EventHandler value
);
public void add_PreSendRequestHeaders(
   System.EventHandler value
);
public void remove_PreSendRequestHeaders(
   System.EventHandler value
);
public void add_ReleaseRequestState(
   System.EventHandler value
);
public void remove_ReleaseRequestState(
   System.EventHandler value
);
public void add_ResolveRequestCache(
   System.EventHandler value
);
public void remove_ResolveRequestCache(
   System.EventHandler value
);
public void add_UpdateRequestCache(
   System.EventHandler value
);
public void remove_UpdateRequestCache(
   System.EventHandler value
);
```

Protected Properties

```
protected System.ComponentModel.EventHandlerList get_Events();
```

HttpApplicationState Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HttpApplicationState
   extends System.Collections.Specialized.NameObjectCollectionBase
```

Public Properties

```
public System.String[] get_AllKeys();
public System.Web.HttpApplicationState get_Contents();
public int get_Count();
public System.Object get_Item(
   int index
);
public System.Object get_Item(
   System.String name
);
public void set_Item(
   System.String name,
   System.Object value
);
public System.Web.HttpStaticObjectsCollection get_StaticObjects();
```

Public Methods

```
public void Add(
   System.String name,
   System.Object value
);
public void Clear();
public System.Object Get(
   int index
);
public System.Object Get(
   System.String name
);
public System.String GetKey(
   int index
);
public void Lock();
public void Remove(
   System.String name
);
public void RemoveAll();
public void RemoveAt(
   int index
);
public void Set(
   System.String name,
   System.Object value
);
public void UnLock();
```

HttpBrowserCapabilities Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpBrowserCapabilities
   extends System.Web.Configuration.HttpCapabilitiesBase
```

Public Constructors

```
public HttpBrowserCapabilities();
```

Public Properties

```
public boolean get_ActiveXControls();
public boolean get_AOL();
public boolean get_BackgroundSounds();
public boolean get_Beta();
public System.String get_Browser();
public boolean get_CDF();
public System.Version get_ClrVersion();
public boolean get_Cookies();
public boolean get_Crawler();
public System.Version get_EcmaScriptVersion();
public boolean get_Frames();
public boolean get_JavaApplets();
public boolean get_JavaScript();
public int get_MajorVersion();
public double get_MinorVersion();
public System.Version get_MSDomVersion();
public System.String get_Platform();
public boolean get_Tables();
public System.Type get_TagWriter();
public System.String get_Type();
public boolean get_VBScript();
public System.String get_Version();
public System.Version get_W3CDomVersion();
public boolean get_Win16();
public boolean get_Win32();
```

HttpCacheability Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum HttpCacheability
```

Members

NoCache

Private

Public

Server

HttpCachePolicy Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpCachePolicy
   extends System.Object
```

Public Properties

```
public System.Web.HttpCacheVaryByHeaders get_VaryByHeaders();
public System.Web.HttpCacheVaryByParams get_VaryByParams();
```

Public Methods

```
public void AddValidationCallback(
   System.Web.HttpCacheValidateHandler handler,
   System.Object data
);
public void AppendCacheExtension(
   System.String extension
);
public void SetCacheability(
   System.Web.HttpCacheability cacheability
);
public void SetCacheability(
   System.Web.HttpCacheability cacheability,
   System.String field
);
public void SetETag(
   System.String etag
);
public void SetETagFromFileDependencies();
public void SetExpires(
   System.DateTime date
);
public void SetLastModified(
   System.DateTime date
);
public void SetLastModifiedFromFileDependencies();
public void SetMaxAge(
   System.TimeSpan delta
);
public void SetNoServerCaching();
public void SetNoStore();
public void SetNoTransforms();
public void SetProxyMaxAge(
   System.TimeSpan delta
);
public void SetRevalidation(
   System.Web.HttpCacheRevalidation revalidation
);
public void SetSlidingExpiration(
   boolean slide
);
public void SetValidUntilExpires(
   boolean validUntilExpires
);
public void SetVaryByCustom(
   System.String custom
```

```
);
```

HttpCacheRevalidation Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum HttpCacheRevalidation
```

Members

AllCaches

None

ProxyCaches

HttpCacheValidateHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void HttpCacheValidateHandler(
   System.Web.HttpContext context,
   System.Object data,
   System.Web.HttpValidationStatus& validationStatus
);
```

HttpCacheVaryByHeaders Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HttpCacheVaryByHeaders
   extends System.Object
```

Public Properties

```
public boolean get_AcceptTypes();
public void set_AcceptTypes(
   boolean value
);
public boolean get_Item(
   System.String header
);
public void set_Item(
   System.String header,
   boolean value
);
public boolean get_UserAgent();
public void set_UserAgent(
   boolean value
);
public boolean get_UserCharSet();
public void set_UserCharSet(
   boolean value
);
public boolean get_UserLanguage();
public void set_UserLanguage(
   boolean value
);
```

Public Methods

```
public void VaryByUnspecifiedParameters();
```

HttpCacheVaryByParams Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HttpCacheVaryByParams
   extends System.Object
```

Public Properties

```
public boolean get_IgnoreParams();
public void set_IgnoreParams(
   boolean value
);
public boolean get_Item(
   System.String header
);
public void set_Item(
   System.String header,
   boolean value
);
```

HttpClientCertificate Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpClientCertificate
   extends System.Collections.Specialized.NameValueCollection
```

Public Properties

```
public ubyte[] get_BinaryIssuer();
public int get_CertEncoding();
public ubyte[] get_Certificate();
public System.String get_Cookie();
public int get_Flags();
public boolean get_IsPresent();
public System.String get_Issuer();
public boolean get_IsValid();
public int get_KeySize();
public ubyte[] get_PublicKey();
public int get_SecretKeySize();
public System.String get_SerialNumber();
public System.String get_ServerIssuer();
public System.String get_ServerSubject();
public System.String get_Subject();
public System.DateTime get_ValidFrom();
public System.DateTime get_ValidUntil();
```

Public Methods

```
public System.String Get(
   System.String field
);
```

HttpCompileException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpCompileException
   extends System.Web.HttpException
```

Public Properties

```
public System.CodeDom.Compiler.CompilerResults get_Results();
public System.String get_SourceCode();
```

HttpContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpContext
   extends System.Object
   implements System.IServiceProvider
```

Public Constructors

```
public HttpContext(
   System.Web.HttpWorkerRequest wr
);
public HttpContext(
   System.Web.HttpRequest request,
   System.Web.HttpResponse response
);
```

Public Properties

```
public System.Exception[] get_AllErrors();
public System.Web.HttpApplicationState get_Application();
public System.Web.HttpApplication get_ApplicationInstance();
public void set_ApplicationInstance(
   System.Web.HttpApplication value
);
public System.Web.Caching.Cache get_Cache();
public static System.Web.HttpContext get_Current();
public System.Exception get_Error();
public System.Web.IHttpHandler get_Handler();
public void set_Handler(
   System.Web.IHttpHandler value
);
public boolean get_IsCustomErrorEnabled();
public boolean get_IsDebuggingEnabled();
public System.Collections.IDictionary get_Items();
public System.Web.HttpRequest get_Request();
public System.Web.HttpResponse get_Response();
public System.Web.HttpServerUtility get_Server();
public System.Web.SessionState.HttpSessionState get_Session();
public boolean get_SkipAuthorization();
public void set_SkipAuthorization(
   boolean value
);
public System.DateTime get_Timestamp();
public System.Web.TraceContext get_Trace();
public System.Security.Principal.IPrincipal get_User();
public void set_User(
   System.Security.Principal.IPrincipal value
);
```

Public Methods

```
public void AddError(
   System.Exception errorInfo
);
public void ClearError();
```

```
public static System.Object GetAppConfig(
   System.String name
);
public System.Object GetConfig(
   System.String name
);
public void RewritePath(
   System.String path
);
```

HttpCookie Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HttpCookie
   extends System.Object
```

Public Constructors

```
public HttpCookie(
   System.String name
);
public HttpCookie(
   System.String name,
   System.String value
);
```

Public Properties

```
public System.String get_Domain();
public void set_Domain(
   System.String value
);
public System.DateTime get_Expires();
public void set_Expires(
   System.DateTime value
);
public boolean get_HasKeys();
public System.String get_Item(
   System.String key
);
public void set_Item(
   System.String key,
   System.String value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_Path();
public void set_Path(
   System.String value
);
public boolean get_Secure();
public void set_Secure(
   boolean value
);
public System.String get_Value();
public void set_Value(
   System.String value
);
public System.Collections.Specialized.NameValueCollection get_Values();
```

HttpCookieCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HttpCookieCollection
   extends System.Collections.Specialized.NameObjectCollectionBase
```

Public Constructors

```
public HttpCookieCollection();
```

Public Properties

```
public System.String[] get_AllKeys();
public System.Web.HttpCookie get_Item(
   int index
);
public System.Web.HttpCookie get_Item(
   System.String name
);
```

Public Methods

```
public void Add(
   System.Web.HttpCookie cookie
);
public void Clear();
public void CopyTo(
   System.Array dest,
   int index
);
public System.Web.HttpCookie Get(
   int index
);
public System.Web.HttpCookie Get(
   System.String name
);
public System.String GetKey(
   int index
);
public void Remove(
   System.String name
);
public void Set(
   System.Web.HttpCookie cookie
);
```

HttpException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpException
   extends System.Runtime.InteropServices.ExternalException
```

Public Constructors

```
public HttpException();
public HttpException(
   System.String message
);
public HttpException(
   int httpCode,
   System.String message
);
public HttpException(
   System.String message,
   int hr
);
public HttpException(
   System.String message,
   System.Exception innerException
);
public HttpException(
   int httpCode,
   System.String message,
   int hr
);
public HttpException(
   int httpCode,
   System.String message,
   System.Exception innerException
);
```

Public Methods

```
public static System.Web.HttpException CreateFromLastError(
   System.String message
);
public System.String GetHtmlErrorMessage();
public int GetHttpCode();
```

HttpFileCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HttpFileCollection
   extends System.Collections.Specialized.NameObjectCollectionBase
```

Public Properties

```
public System.String[] get_AllKeys();
public System.Web.HttpPostedFile get_Item(
   int index
);
public System.Web.HttpPostedFile get_Item(
   System.String name
);
```

Public Methods

```
public void CopyTo(
   System.Array dest,
   int index
);
public System.Web.HttpPostedFile Get(
   int index
);
public System.Web.HttpPostedFile Get(
   System.String name
);
public System.String GetKey(
   int index
);
```

HttpModuleCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HttpModuleCollection
   extends System.Collections.Specialized.NameObjectCollectionBase
```

Public Properties

```
public System.String[] get_AllKeys();
public System.Web.IHttpModule get_Item(
   int index
);
public System.Web.IHttpModule get_Item(
   System.String name
);
```

Public Methods

```
public void CopyTo(
   System.Array dest,
   int index
);
public System.Web.IHttpModule Get(
   int index
);
public System.Web.IHttpModule Get(
   System.String name
);
public System.String GetKey(
   int index
);
```

HttpParseException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpParseException
   extends System.Web.HttpException
```

Public Properties

```
public System.String get_FileName();
public int get_Line();
```

HttpPostedFile Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpPostedFile
   extends System.Object
```

Public Properties

```
public int get_ContentLength();
public System.String get_ContentType();
public System.String get_FileName();
public System.IO.Stream get_InputStream();
```

Public Methods

```
public void SaveAs(
   System.String filename
);
```

HttpRequest Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HttpRequest
   extends System.Object
```

Public Constructors

```
public HttpRequest(
   System.String filename,
   System.String url,
   System.String queryString
);
```

Public Properties

```
public System.String[] get_AcceptTypes();
public System.String get_ApplicationPath();
public System.Web.HttpBrowserCapabilities get_Browser();
public void set_Browser(
   System.Web.HttpBrowserCapabilities value
);
public System.Web.HttpClientCertificate get_ClientCertificate();
public System.Text.Encoding get_ContentEncoding();
public void set_ContentEncoding(
   System.Text.Encoding value
);
public int get_ContentLength();
public System.String get_ContentType();
public System.Web.HttpCookieCollection get_Cookies();
public System.String get_CurrentExecutionFilePath();
public System.String get_FilePath();
public System.Web.HttpFileCollection get_Files();
public System.IO.Stream get_Filter();
public void set_Filter(
   System.IO.Stream value
);
public System.Collections.Specialized.NameValueCollection get_Form();
public System.Collections.Specialized.NameValueCollection get_Headers();
public System.String get_HttpMethod();
public System.IO.Stream get_InputStream();
public boolean get_IsAuthenticated();
public boolean get_IsSecureConnection();
public System.String get_Item(
   System.String key
);
public System.Collections.Specialized.NameValueCollection get_Params();
public System.String get_Path();
public System.String get_PathInfo();
public System.String get_PhysicalApplicationPath();
public System.String get_PhysicalPath();
public System.Collections.Specialized.NameValueCollection get_QueryString();
public System.String get_RawUrl();
public System.String get_RequestType();
public void set_RequestType(
   System.String value
);
```

```
public System.Collections.Specialized.NameValueCollection get_ServerVariables();
public int get_TotalBytes();
public System.Uri get_Url();
public System.Uri get_UrlReferrer();
public System.String get_UserAgent();
public System.String get_UserHostAddress();
public System.String get_UserHostName();
public System.String[] get_UserLanguages();
```

Public Methods

```
public ubyte[] BinaryRead(
   int count
);
public int[] MapImageCoordinates(
   System.String imageFieldName
);
public System.String MapPath(
   System.String virtualPath
);
public System.String MapPath(
   System.String virtualPath,
   System.String baseVirtualDir,
   boolean allowCrossAppMapping
);
public void SaveAs(
   System.String filename,
   boolean includeHeaders
);
```

HttpResponse Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpResponse
   extends System.Object
```

Public Constructors

```
public HttpResponse(
   System.IO.TextWriter writer
);
```

Public Properties

```
public boolean get_Buffer();
public void set_Buffer(
   boolean value
);
public boolean get_BufferOutput();
public void set_BufferOutput(
   boolean value
);
public System.Web.HttpCachePolicy get_Cache();
public System.String get_CacheControl();
public void set_CacheControl(
   System.String value
);
public System.String get_Charset();
public void set_Charset(
   System.String value
);
public System.Text.Encoding get_ContentEncoding();
public void set_ContentEncoding(
   System.Text.Encoding value
);
public System.String get_ContentType();
public void set_ContentType(
   System.String value
);
public System.Web.HttpCookieCollection get_Cookies();
public int get_Expires();
public void set_Expires(
   int value
);
public System.DateTime get_ExpiresAbsolute();
public void set_ExpiresAbsolute(
   System.DateTime value
);
public System.IO.Stream get_Filter();
public void set_Filter(
   System.IO.Stream value
);
public boolean get_IsClientConnected();
public System.IO.TextWriter get_Output();
public System.IO.Stream get_OutputStream();
public System.String get_Status();
public void set_Status(
   System.String value
```

```
);
public int get_StatusCode();
public void set_StatusCode(
   int value
);
public System.String get_StatusDescription();
public void set_StatusDescription(
   System.String value
);
public boolean get_SuppressContent();
public void set_SuppressContent(
   boolean value
);
```

Public Methods

```
public void AddCacheItemDependencies(
   System.Collections.ArrayList cacheKeys
);
public void AddCacheItemDependency(
   System.String cacheKey
);
public void AddFileDependencies(
   System.Collections.ArrayList filenames
);
public void AddFileDependency(
   System.String filename
);
public void AddHeader(
   System.String name,
   System.String value
);
public void AppendCookie(
   System.Web.HttpCookie cookie
);
public void AppendHeader(
   System.String name,
   System.String value
);
public void AppendToLog(
   System.String param
);
public System.String ApplyAppPathModifier(
   System.String virtualPath
);
public void BinaryWrite(
   ubyte[] buffer
);
public void Clear();
public void ClearContent();
public void ClearHeaders();
public void Close();
public void End();
public void Flush();
public void Pics(
   System.String value
);
public void Redirect(
   System.String url
);
public void Redirect(
   System.String url,
   boolean endResponse
);
public static void RemoveOutputCacheItem(
   System.String path
```

```
);
public void SetCookie(
   System.Web.HttpCookie cookie
);
public void Write(
   char ch
);
public void Write(
   System.Object obj
);
public void Write(
   System.String s
);
public void Write(
   char[] buffer,
   int index,
   int count
);
public void WriteFile(
   System.String filename
);
public void WriteFile(
   System.String filename,
   boolean readIntoMemory
);
public void WriteFile(
   System.IntPtr fileHandle,
   long offset,
   long size
);
public void WriteFile(
   System.String filename,
   long offset,
   long size
);
```

HttpRuntime Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpRuntime
   extends System.Object
```

Public Constructors

```
public HttpRuntime();
```

Public Properties

```
public static System.String get_AppDomainAppId();
public static System.String get_AppDomainAppPath();
public static System.String get_AppDomainAppVirtualPath();
public static System.String get_AppDomainId();
public static System.String get_AspInstallDirectory();
public static System.String get_BinDirectory();
public static System.Web.Caching.Cache get_Cache();
public static System.String get_ClrInstallDirectory();
public static System.String get_CodegenDir();
public static boolean get_IsOnUNCShare();
public static System.String get_MachineConfigurationDirectory();
```

Public Methods

```
public static void Close();
public static void ProcessRequest(
   System.Web.HttpWorkerRequest wr
);
```

HttpServerUtility Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpServerUtility
   extends System.Object
```

Public Properties

```
public System.String get_MachineName();
public int get_ScriptTimeout();
public void set_ScriptTimeout(
   int value
);
```

Public Methods

```
public void ClearError();
public System.Object CreateObject(
   System.String progID
);
public System.Object CreateObject(
   System.Type type
);
public System.Object CreateObjectFromClsid(
   System.String clsid
);
public void Execute(
   System.String path
);
public void Execute(
   System.String path,
   System.IO.TextWriter writer
);
public System.Exception GetLastError();
public System.String HtmlDecode(
   System.String s
);
public void HtmlDecode(
   System.String s,
   System.IO.TextWriter output
);
public System.String HtmlEncode(
   System.String s
);
public void HtmlEncode(
   System.String s,
   System.IO.TextWriter output
);
public System.String MapPath(
   System.String path
);
public void Transfer(
   System.String path
);
public void Transfer(
   System.String path,
   boolean preserveForm
);
```

```
public System.String UrlDecode(
   System.String s
);
public void UrlDecode(
   System.String s,
   System.IO.TextWriter output
);
public System.String UrlEncode(
   System.String s
);
public void UrlEncode(
   System.String s,
   System.IO.TextWriter output
);
public System.String UrlPathEncode(
   System.String s
);
```

HttpStaticObjectsCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HttpStaticObjectsCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public HttpStaticObjectsCollection();
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Object get_Item(
   System.String name
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public System.Object GetObject(
   System.String name
);
```

HttpUnhandledException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpUnhandledException
   extends System.Web.HttpException
```

HttpUtility Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpUtility
   extends System.Object
```

Public Constructors

```
public HttpUtility();
```

Public Methods

```
public static System.String HtmlAttributeEncode(
   System.String s
);
public static void HtmlAttributeEncode(
   System.String s,
   System.IO.TextWriter output
);
public static System.String HtmlDecode(
   System.String s
);
public static void HtmlDecode(
   System.String s,
   System.IO.TextWriter output
);
public static System.String HtmlEncode(
   System.String s
);
public static void HtmlEncode(
   System.String s,
   System.IO.TextWriter output
);
public static System.String UrlDecode(
   System.String str
);
public static System.String UrlDecode(
   ubyte[] bytes,
   System.Text.Encoding e
);
public static System.String UrlDecode(
   System.String str,
   System.Text.Encoding e
);
public static System.String UrlDecode(
   ubyte[] bytes,
   int offset,
   int count,
   System.Text.Encoding e
);
public static ubyte[] UrlDecodeToBytes(
   ubyte[] bytes
);
public static ubyte[] UrlDecodeToBytes(
   System.String str
);
public static ubyte[] UrlDecodeToBytes(
   System.String str,
```

```
   System.Text.Encoding e
);
public static ubyte[] UrlDecodeToBytes(
   ubyte[] bytes,
   int offset,
   int count
);
public static System.String UrlEncode(
   ubyte[] bytes
);
public static System.String UrlEncode(
   System.String str
);
public static System.String UrlEncode(
   System.String str,
   System.Text.Encoding e
);
public static System.String UrlEncode(
   ubyte[] bytes,
   int offset,
   int count
);
public static ubyte[] UrlEncodeToBytes(
   ubyte[] bytes
);
public static ubyte[] UrlEncodeToBytes(
   System.String str
);
public static ubyte[] UrlEncodeToBytes(
   System.String str,
   System.Text.Encoding e
);
public static ubyte[] UrlEncodeToBytes(
   ubyte[] bytes,
   int offset,
   int count
);
public static System.String UrlEncodeUnicode(
   System.String str
);
public static ubyte[] UrlEncodeUnicodeToBytes(
   System.String str
);
```

HttpValidationStatus Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum HttpValidationStatus
```

Members

IgnoreThisRequest

Invalid

Valid

HttpWorkerRequest Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public abstract class HttpWorkerRequest
   extends System.Object
   implements System.Web.IHttpMapPath
```

Public Properties

```
public System.String get_MachineConfigPath();
public System.String get_MachineInstallDirectory();
```

Public Methods

```
public void CloseConnection();
public abstract void EndOfRequest();
public abstract void FlushResponse(
   boolean finalFlush
);
public System.String GetAppPath();
public System.String GetAppPathTranslated();
public System.String GetAppPoolID();
public long GetBytesRead();
public ubyte[] GetClientCertificate();
public ubyte[] GetClientCertificateBinaryIssuer();
public int GetClientCertificateEncoding();
public ubyte[] GetClientCertificatePublicKey();
public System.DateTime GetClientCertificateValidFrom();
public System.DateTime GetClientCertificateValidUntil();
public long GetConnectionID();
public System.String GetFilePath();
public System.String GetFilePathTranslated();
public abstract System.String GetHttpVerbName();
public abstract System.String GetHttpVersion();
public System.String GetKnownRequestHeader(
   int index
);
public static int GetKnownRequestHeaderIndex(
   System.String header
);
public static System.String GetKnownRequestHeaderName(
   int index
);
public static int GetKnownResponseHeaderIndex(
   System.String header
);
public static System.String GetKnownResponseHeaderName(
   int index
);
public abstract System.String GetLocalAddress();
public abstract int GetLocalPort();
public System.String GetPathInfo();
public ubyte[] GetPreloadedEntityBody();
public System.String GetProtocol();
public abstract System.String GetQueryString();
public ubyte[] GetQueryStringRawBytes();
public abstract System.String GetRawUrl();
```

```
public abstract System.String GetRemoteAddress();
public System.String GetRemoteName();
public abstract int GetRemotePort();
public int GetRequestReason();
public System.String GetServerName();
public System.String GetServerVariable(
   System.String name
);
public static System.String GetStatusDescription(
   int code
);
public System.String GetUnknownRequestHeader(
   System.String name
);
public abstract System.String GetUriPath();
public long GetUrlContextID();
public System.IntPtr GetUserToken();
public System.IntPtr GetVirtualPathToken();
public boolean HasEntityBody();
public boolean HeadersSent();
public boolean IsClientConnected();
public boolean IsEntireEntityBodyIsPreloaded();
public boolean IsSecure();
public System.String MapPath(
   System.String virtualPath
);
public int ReadEntityBody(
   ubyte[] buffer,
   int size
);
public void SendCalculatedContentLength(
   int contentLength
);
public abstract void SendKnownResponseHeader(
   int index,
   System.String value
);
public abstract void SendResponseFromFile(
   System.IntPtr handle,
   long offset,
   long length
);
public abstract void SendResponseFromFile(
   System.String filename,
   long offset,
   long length
);
public abstract void SendResponseFromMemory(
   ubyte[] data,
   int length
);
public void SendResponseFromMemory(
   System.IntPtr data,
   int length
);
public abstract void SendStatus(
   int statusCode,
   System.String statusDescription
);
public abstract void SendUnknownResponseHeader(
   System.String name,
   System.String value
);
public void SetEndOfSendNotification(
   System.Web.HttpWorkerRequest+EndOfSendNotification callback,
   System.Object extraData
);
```

Public Fields

```
public static int HeaderAccept;
public static int HeaderAcceptCharset;
public static int HeaderAcceptEncoding;
public static int HeaderAcceptLanguage;
public static int HeaderAcceptRanges;
public static int HeaderAge;
public static int HeaderAllow;
public static int HeaderAuthorization;
public static int HeaderCacheControl;
public static int HeaderConnection;
public static int HeaderContentEncoding;
public static int HeaderContentLanguage;
public static int HeaderContentLength;
public static int HeaderContentLocation;
public static int HeaderContentMd5;
public static int HeaderContentRange;
public static int HeaderContentType;
public static int HeaderCookie;
public static int HeaderDate;
public static int HeaderEtag;
public static int HeaderExpect;
public static int HeaderExpires;
public static int HeaderFrom;
public static int HeaderHost;
public static int HeaderIfMatch;
public static int HeaderIfModifiedSince;
public static int HeaderIfNoneMatch;
public static int HeaderIfRange;
public static int HeaderIfUnmodifiedSince;
public static int HeaderKeepAlive;
public static int HeaderLastModified;
public static int HeaderLocation;
public static int HeaderMaxForwards;
public static int HeaderPragma;
public static int HeaderProxyAuthenticate;
public static int HeaderProxyAuthorization;
public static int HeaderRange;
public static int HeaderReferer;
public static int HeaderRetryAfter;
public static int HeaderServer;
public static int HeaderSetCookie;
public static int HeaderTe;
public static int HeaderTrailer;
public static int HeaderTransferEncoding;
public static int HeaderUpgrade;
public static int HeaderUserAgent;
public static int HeaderVary;
public static int HeaderVia;
public static int HeaderWarning;
public static int HeaderWwwAuthenticate;
public static int ReasonCachePolicy;
public static int ReasonCacheSecurity;
public static int ReasonClientDisconnect;
public static int ReasonDefault;
public static int ReasonFileHandleCacheMiss;
public static int ReasonResponseCacheMiss;
public static int RequestHeaderMaximum;
public static int ResponseHeaderMaximum;
```

Protected Constructors

```
protected HttpWorkerRequest();
```

HttpWorkerRequest.EndOfSendNotification Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void EndOfSendNotification(
   System.Web.HttpWorkerRequest wr,
   System.Object extraData
);
```

HttpWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpWriter
   extends System.IO.TextWriter
```

Public Properties

```
public System.Text.Encoding get_Encoding();
public System.IO.Stream get_OutputStream();
```

Public Methods

```
public void Close();
public void Flush();
public void Write(
   char ch
);
public void Write(
   System.Object obj
);
public void Write(
   System.String s
);
public void Write(
   char[] buffer,
   int index,
   int count
);
public void WriteBytes(
   ubyte[] buffer,
   int index,
   int count
);
public void WriteLine();
public void WriteString(
   System.String s,
   int index,
   int count
);
```

IHttpAsyncHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IHttpAsyncHandler
   implements System.Web.IHttpHandler
```

Public Methods

```
System.IAsyncResult BeginProcessRequest(
   System.Web.HttpContext context,
   System.AsyncCallback cb,
   System.Object extraData
);
void EndProcessRequest(
   System.IAsyncResult result
);
```

IHttpHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IHttpHandler
```

Public Properties

```
boolean get_IsReusable();
```

Public Methods

```
void ProcessRequest(
   System.Web.HttpContext context
);
```

IHttpHandlerFactory Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IHttpHandlerFactory
```

Public Methods

```
System.Web.IHttpHandler GetHandler(
   System.Web.HttpContext context,
   System.String requestType,
   System.String url,
   System.String pathTranslated
);
void ReleaseHandler(
   System.Web.IHttpHandler handler
);
```

IHttpModule Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IHttpModule
```

Public Methods

```
void Dispose();
void Init(
   System.Web.HttpApplication context
);
```

ProcessInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ProcessInfo
   extends System.Object
```

Public Constructors

```
public ProcessInfo();
public ProcessInfo(
   System.DateTime startTime,
   System.TimeSpan age,
   int processID,
   int requestCount,
   System.Web.ProcessStatus status,
   System.Web.ProcessShutdownReason shutdownReason,
   int peakMemoryUsed
);
```

Public Properties

```
public System.TimeSpan get_Age();
public int get_PeakMemoryUsed();
public int get_ProcessID();
public int get_RequestCount();
public System.Web.ProcessShutdownReason get_ShutdownReason();
public System.DateTime get_StartTime();
public System.Web.ProcessStatus get_Status();
```

Public Methods

```
public void SetAll(
   System.DateTime startTime,
   System.TimeSpan age,
   int processID,
   int requestCount,
   System.Web.ProcessStatus status,
   System.Web.ProcessShutdownReason shutdownReason,
   int peakMemoryUsed
);
```

ProcessModelInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ProcessModelInfo
   extends System.Object
```

Public Constructors

```
public ProcessModelInfo();
```

Public Methods

```
public static System.Web.ProcessInfo GetCurrentProcessInfo();
public static System.Web.ProcessInfo[] GetHistory(
   int numRecords
);
```

ProcessShutdownReason Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ProcessShutdownReason
```

Members

DeadlockSuspected

IdleTimeout

MemoryLimitExceeded

None

PingFailed

RequestQueueLimit

RequestsLimit

Timeout

Unexpected

ProcessStatus Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ProcessStatus
```

Members

Alive

ShutDown

ShuttingDown

Terminated

TraceContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TraceContext
   extends System.Object
```

Public Constructors

```
public TraceContext(
   System.Web.HttpContext context
);
```

Public Properties

```
public boolean get_IsEnabled();
public void set_IsEnabled(
   boolean value
);
public System.Web.TraceMode get_TraceMode();
public void set_TraceMode(
   System.Web.TraceMode value
);
```

Public Methods

```
public void Warn(
   System.String message
);
public void Warn(
   System.String category,
   System.String message
);
public void Warn(
   System.String category,
   System.String message,
   System.Exception errorInfo
);
public void Write(
   System.String message
);
public void Write(
   System.String category,
   System.String message
);
public void Write(
   System.String category,
   System.String message,
   System.Exception errorInfo
);
```

TraceMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TraceMode
```

Members

Default

SortByCategory

SortByTime

System.Web.Caching Namespace (Visual J# Syntax)

Classes

Cache

CacheDependency

Delegates

CacheItemRemovedCallback

Enumerations

CacheItemPriority

CacheItemRemovedReason

Cache Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class Cache
   extends System.Object
   implements System.Collections.IEnumerable
```

Public Constructors

```
public Cache();
```

Public Properties

```
public int get_Count();
public System.Object get_Item(
   System.String key
);
public void set_Item(
   System.String key,
   System.Object value
);
```

Public Methods

```
public System.Object Add(
   System.String key,
   System.Object value,
   System.Web.Caching.CacheDependency dependencies,
   System.DateTime absoluteExpiration,
   System.TimeSpan slidingExpiration,
   System.Web.Caching.CacheItemPriority priority,
   System.Web.Caching.CacheItemRemovedCallback onRemoveCallback
);
public System.Object Get(
   System.String key
);
public System.Collections.IDictionaryEnumerator GetEnumerator();
public void Insert(
   System.String key,
   System.Object value
);
public void Insert(
   System.String key,
   System.Object value,
   System.Web.Caching.CacheDependency dependencies
);
public void Insert(
   System.String key,
   System.Object value,
   System.Web.Caching.CacheDependency dependencies,
   System.DateTime absoluteExpiration,
   System.TimeSpan slidingExpiration
);
public void Insert(
   System.String key,
```

```
    System.Object value,
    System.Web.Caching.CacheDependency dependencies,
    System.DateTime absoluteExpiration,
    System.TimeSpan slidingExpiration,
    System.Web.Caching.CacheItemPriority priority,
    System.Web.Caching.CacheItemRemovedCallback onRemoveCallback
);
public System.Object Remove(
    System.String key
);
```

Public Fields

```
public static final System.DateTime NoAbsoluteExpiration;
public static final System.TimeSpan NoSlidingExpiration;
```

CacheDependency Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CacheDependency
   extends System.Object
   implements System.IDisposable
```

Public Constructors

```
public CacheDependency(
   System.String filename
);
public CacheDependency(
   System.String[] filenames
);
public CacheDependency(
   System.String filename,
   System.DateTime start
);
public CacheDependency(
   System.String[] filenames,
   System.DateTime start
);
public CacheDependency(
   System.String[] filenames,
   System.String[] cachekeys
);
public CacheDependency(
   System.String[] filenames,
   System.String[] cachekeys,
   System.DateTime start
);
public CacheDependency(
   System.String[] filenames,
   System.String[] cachekeys,
   System.Web.Caching.CacheDependency dependency
);
public CacheDependency(
   System.String[] filenames,
   System.String[] cachekeys,
   System.Web.Caching.CacheDependency dependency,
   System.DateTime start
);
```

Public Properties

```
public boolean get_HasChanged();
```

Public Methods

```
public final void Dispose();
```

CacheItemPriority Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CacheItemPriority
```

Members

AboveNormal

BelowNormal

Default

High

Low

Normal

NotRemovable

CacheItemRemovedCallback Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void CacheItemRemovedCallback(
   System.String key,
   System.Object value,
   System.Web.Caching.CacheItemRemovedReason reason
);
```

CacheItemRemovedReason Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CacheItemRemovedReason
```

Members

DependencyChanged

Expired

Removed

Underused

System.Web.Configuration Namespace (Visual J# Syntax)

Classes

HttpCapabilitiesBase

HttpConfigurationContext

Enumerations

AuthenticationMode

FormsAuthPasswordFormat

FormsProtectionEnum

AuthenticationMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum AuthenticationMode
```

Members

Forms

None

Passport

Windows

FormsAuthPasswordFormat Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum FormsAuthPasswordFormat
```

Members

Clear

MD5

SHA1

FormsProtectionEnum Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum FormsProtectionEnum
```

Members

All

Encryption

None

Validation

HttpCapabilitiesBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HttpCapabilitiesBase
   extends System.Object
```

Public Constructors

```
public HttpCapabilitiesBase();
```

Public Properties

```
public System.String get_Item(
   System.String key
);
```

Public Methods

```
public static System.Web.Configuration.HttpCapabilitiesBase GetConfigCapabilities(
   System.String configKey,
   System.Web.HttpRequest request
);
```

Protected Methods

```
protected void Init();
```

HttpConfigurationContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpConfigurationContext
   extends System.Object
```

Public Properties

```
public System.String get_VirtualPath();
```

System.Web.Hosting Namespace (Visual J# Syntax)

Classes

AppDomainFactory

ApplicationHost

ISAPIRuntime

SimpleWorkerRequest

Interfaces

IAppDomainFactory

IISAPIRuntime

AppDomainFactory Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AppDomainFactory
   extends System.Object
   implements System.Web.Hosting.IAppDomainFactory
```

Public Constructors

```
public AppDomainFactory();
```

Public Methods

```
public final System.Object Create(
   System.String module,
   System.String typeName,
   System.String appId,
   System.String appPath,
   System.String strUrlOfAppOrigin,
   int iZone
);
```

ApplicationHost Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ApplicationHost
   extends System.Object
```

Public Methods

```
public static System.Object CreateApplicationHost(
   System.Type hostType,
   System.String virtualDir,
   System.String physicalDir
);
```

IAppDomainFactory Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "e6e21054-a7dc-4378-877d-
b7f4a2d7e8ba") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface IAppDomainFactory
```

Public Methods

```
System.Object Create(
   System.String module,
   System.String typeName,
   System.String appId,
   System.String appPath,
   System.String strUrlOfAppOrigin,
   int iZone
);
```

IISAPIRuntime Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "08a2c56f-7c16-41c1-a8be-
432917a1a2d1") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface IISAPIRuntime
```

Public Methods

```
void DoGCCollect();
int ProcessRequest(
   System.IntPtr ecb,
   int useProcessModel
);
void StartProcessing();
void StopProcessing();
```

ISAPIRuntime Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ISAPIRuntime
   extends System.Object
   implements System.Web.Hosting.IISAPIRuntime
```

Public Constructors

```
public ISAPIRuntime();
```

Public Methods

```
public final void DoGCCollect();
public final int ProcessRequest(
   System.IntPtr ecb,
   int iWRType
);
public final void StartProcessing();
public final void StopProcessing();
```

SimpleWorkerRequest Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class SimpleWorkerRequest
   extends System.Web.HttpWorkerRequest
```

Public Constructors

```
public SimpleWorkerRequest(
   System.String page,
   System.String query,
   System.IO.TextWriter output
);
public SimpleWorkerRequest(
   System.String appVirtualDir,
   System.String appPhysicalDir,
   System.String page,
   System.String query,
   System.IO.TextWriter output
);
```

Public Properties

```
public System.String get_MachineConfigPath();
public System.String get_MachineInstallDirectory();
```

Public Methods

```
public void EndOfRequest();
public void FlushResponse(
   boolean finalFlush
);
public System.String GetAppPath();
public System.String GetAppPathTranslated();
public System.String GetFilePath();
public System.String GetFilePathTranslated();
public System.String GetHttpVerbName();
public System.String GetHttpVersion();
public System.String GetLocalAddress();
public int GetLocalPort();
public System.String GetPathInfo();
public System.String GetQueryString();
public System.String GetRawUrl();
public System.String GetRemoteAddress();
public int GetRemotePort();
public System.String GetServerVariable(
   System.String name
);
public System.String GetUriPath();
public System.IntPtr GetUserToken();
public System.String MapPath(
   System.String path
);
public void SendKnownResponseHeader(
   int index,
```

```
      System.String value
   );
   public void SendResponseFromFile(
      System.IntPtr handle,
      long offset,
      long length
   );
   public void SendResponseFromFile(
      System.String filename,
      long offset,
      long length
   );
   public void SendResponseFromMemory(
      ubyte[] data,
      int length
   );
   public void SendStatus(
      int statusCode,
      System.String statusDescription
   );
   public void SendUnknownResponseHeader(
      System.String name,
      System.String value
   );
```

System.Web.Mail Namespace (Visual J# Syntax)

Classes

MailAttachment

MailMessage

SmtpMail

Enumerations

MailEncoding

MailFormat

MailPriority

MailAttachment Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MailAttachment
   extends System.Object
```

Public Constructors

```
public MailAttachment(
   System.String filename
);
public MailAttachment(
   System.String filename,
   System.Web.Mail.MailEncoding encoding
);
```

Public Properties

```
public System.Web.Mail.MailEncoding get_Encoding();
public System.String get_Filename();
```

MailEncoding Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MailEncoding
```

Members

Base64

UUEncode

MailFormat Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MailFormat
```

Members

Html

Text

MailMessage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MailMessage
   extends System.Object
```

Public Constructors

```
public MailMessage();
```

Public Properties

```
public System.Collections.IList get_Attachments();
public System.String get_Bcc();
public void set_Bcc(
   System.String value
);
public System.String get_Body();
public void set_Body(
   System.String value
);
public System.Text.Encoding get_BodyEncoding();
public void set_BodyEncoding(
   System.Text.Encoding value
);
public System.Web.Mail.MailFormat get_BodyFormat();
public void set_BodyFormat(
   System.Web.Mail.MailFormat value
);
public System.String get_Cc();
public void set_Cc(
   System.String value
);
public System.String get_From();
public void set_From(
   System.String value
);
public System.Collections.IDictionary get_Headers();
public System.Web.Mail.MailPriority get_Priority();
public void set_Priority(
   System.Web.Mail.MailPriority value
);
public System.String get_Subject();
public void set_Subject(
   System.String value
);
public System.String get_To();
public void set_To(
   System.String value
);
public System.String get_UrlContentBase();
public void set_UrlContentBase(
   System.String value
);
public System.String get_UrlContentLocation();
public void set_UrlContentLocation(
   System.String value
);
```

MailPriority Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MailPriority
```

Members

High

Low

Normal

SmtpMail Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SmtpMail
   extends System.Object
```

Public Properties

```
public static System.String get_SmtpServer();
public static void set_SmtpServer(
   System.String value
);
```

Public Methods

```
public static void Send(
   System.Web.Mail.MailMessage message
);
public static void Send(
   System.String from,
   System.String to,
   System.String subject,
   System.String messageText
);
```

System.Web.Security Namespace (Visual J# Syntax)

Classes

DefaultAuthenticationEventArgs

DefaultAuthenticationModule

FileAuthorizationModule

FormsAuthentication

FormsAuthenticationEventArgs

FormsAuthenticationModule

FormsAuthenticationTicket

FormsIdentity

PassportAuthenticationEventArgs

PassportAuthenticationModule

PassportIdentity

UrlAuthorizationModule

WindowsAuthenticationEventArgs

WindowsAuthenticationModule

Delegates

DefaultAuthenticationEventHandler

FormsAuthenticationEventHandler

PassportAuthenticationEventHandler

WindowsAuthenticationEventHandler

DefaultAuthenticationEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DefaultAuthenticationEventArgs
   extends System.EventArgs
```

Public Constructors

```
public DefaultAuthenticationEventArgs(
   System.Web.HttpContext context
);
```

Public Properties

```
public System.Web.HttpContext get_Context();
```

DefaultAuthenticationEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DefaultAuthenticationEventHandler(
   System.Object sender,
   System.Web.Security.DefaultAuthenticationEventArgs e
);
```

DefaultAuthenticationModule Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DefaultAuthenticationModule
   extends System.Object
   implements System.Web.IHttpModule
```

Public Constructors

```
public DefaultAuthenticationModule();
```

Public Methods

```
public final void Dispose();
public final void Init(
   System.Web.HttpApplication app
);
```

Public Events

```
public void add_Authenticate(
   System.Web.Security.DefaultAuthenticationEventHandler value
);
public void remove_Authenticate(
   System.Web.Security.DefaultAuthenticationEventHandler value
);
```

FileAuthorizationModule Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FileAuthorizationModule
   extends System.Object
   implements System.Web.IHttpModule
```

Public Constructors

```
public FileAuthorizationModule();
```

Public Methods

```
public final void Dispose();
public final void Init(
   System.Web.HttpApplication app
);
```

FormsAuthentication Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FormsAuthentication
   extends System.Object
```

Public Constructors

```
public FormsAuthentication();
```

Public Properties

```
public static System.String get_FormsCookieName();
public static System.String get_FormsCookiePath();
```

Public Methods

```
public static boolean Authenticate(
   System.String name,
   System.String password
);
public static System.Web.Security.FormsAuthenticationTicket Decrypt(
   System.String encryptedTicket
);
public static System.String Encrypt(
   System.Web.Security.FormsAuthenticationTicket ticket
);
public static System.Web.HttpCookie GetAuthCookie(
   System.String userName,
   boolean createPersistentCookie
);
public static System.Web.HttpCookie GetAuthCookie(
   System.String userName,
   boolean createPersistentCookie,
   System.String strCookiePath
);
public static System.String GetRedirectUrl(
   System.String userName,
   boolean createPersistentCookie
);
public static System.String HashPasswordForStoringInConfigFile(
   System.String password,
   System.String passwordFormat
);
public static void Initialize();
public static void RedirectFromLoginPage(
   System.String userName,
   boolean createPersistentCookie
);
public static void RedirectFromLoginPage(
   System.String userName,
   boolean createPersistentCookie,
   System.String strCookiePath
);
public static System.Web.Security.FormsAuthenticationTicket RenewTicketIfOld(
   System.Web.Security.FormsAuthenticationTicket tOld
```

```
);
public static void SetAuthCookie(
   System.String userName,
   boolean createPersistentCookie
);
public static void SetAuthCookie(
   System.String userName,
   boolean createPersistentCookie,
   System.String strCookiePath
);
public static void SignOut();
```

FormsAuthenticationEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FormsAuthenticationEventArgs
   extends System.EventArgs
```

Public Constructors

```
public FormsAuthenticationEventArgs(
   System.Web.HttpContext context
);
```

Public Properties

```
public System.Web.HttpContext get_Context();
public System.Security.Principal.IPrincipal get_User();
public void set_User(
   System.Security.Principal.IPrincipal value
);
```

FormsAuthenticationEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void FormsAuthenticationEventHandler(
   System.Object sender,
   System.Web.Security.FormsAuthenticationEventArgs e
);
```

FormsAuthenticationModule Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FormsAuthenticationModule
   extends System.Object
   implements System.Web.IHttpModule
```

Public Constructors

```
public FormsAuthenticationModule();
```

Public Methods

```
public final void Dispose();
public final void Init(
   System.Web.HttpApplication app
);
```

Public Events

```
public void add_Authenticate(
   System.Web.Security.FormsAuthenticationEventHandler value
);
public void remove_Authenticate(
   System.Web.Security.FormsAuthenticationEventHandler value
);
```

FormsAuthenticationTicket Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class FormsAuthenticationTicket
   extends System.Object
```

Public Constructors

```
public FormsAuthenticationTicket(
   System.String name,
   boolean isPersistent,
   int timeout
);
public FormsAuthenticationTicket(
   int version,
   System.String name,
   System.DateTime issueDate,
   System.DateTime expiration,
   boolean isPersistent,
   System.String userData
);
public FormsAuthenticationTicket(
   int version,
   System.String name,
   System.DateTime issueDate,
   System.DateTime expiration,
   boolean isPersistent,
   System.String userData,
   System.String cookiePath
);
```

Public Properties

```
public System.String get_CookiePath();
public System.DateTime get_Expiration();
public boolean get_Expired();
public boolean get_IsPersistent();
public System.DateTime get_IssueDate();
public System.String get_Name();
public System.String get_UserData();
public int get_Version();
```

FormsIdentity Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class FormsIdentity
   extends System.Object
   implements System.Security.Principal.IIdentity
```

Public Constructors

```
public FormsIdentity(
   System.Web.Security.FormsAuthenticationTicket ticket
);
```

Public Properties

```
public final System.String get_AuthenticationType();
public final boolean get_IsAuthenticated();
public final System.String get_Name();
public System.Web.Security.FormsAuthenticationTicket get_Ticket();
```

PassportAuthenticationEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PassportAuthenticationEventArgs
   extends System.EventArgs
```

Public Constructors

```
public PassportAuthenticationEventArgs(
   System.Web.Security.PassportIdentity identity,
   System.Web.HttpContext context
);
```

Public Properties

```
public System.Web.HttpContext get_Context();
public System.Web.Security.PassportIdentity get_Identity();
public System.Security.Principal.IPrincipal get_User();
public void set_User(
   System.Security.Principal.IPrincipal value
);
```

PassportAuthenticationEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void PassportAuthenticationEventHandler(
   System.Object sender,
   System.Web.Security.PassportAuthenticationEventArgs e
);
```

PassportAuthenticationModule Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PassportAuthenticationModule
   extends System.Object
   implements System.Web.IHttpModule
```

Public Constructors

```
public PassportAuthenticationModule();
```

Public Methods

```
public final void Dispose();
public final void Init(
   System.Web.HttpApplication app
);
```

Public Events

```
public void add_Authenticate(
   System.Web.Security.PassportAuthenticationEventHandler value
);
public void remove_Authenticate(
   System.Web.Security.PassportAuthenticationEventHandler value
);
```

PassportIdentity Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class PassportIdentity
   extends System.Object
   implements System.Security.Principal.IIdentity
```

Public Constructors

```
public PassportIdentity();
```

Public Properties

```
public final System.String get_AuthenticationType();
public int get_Error();
public boolean get_GetFromNetworkServer();
public boolean get_HasSavedPassword();
public boolean get_HasTicket();
public System.String get_HexPUID();
public final boolean get_IsAuthenticated();
public System.String get_Item(
   System.String strProfileName
);
public final System.String get_Name();
public int get_TicketAge();
public int get_TimeSinceSignIn();
```

Public Methods

```
public System.String AuthUrl();
public System.String AuthUrl(
   System.String strReturnUrl,
   int iTimeWindow,
   boolean fForceLogin,
   System.String strCoBrandedArgs,
   int iLangID,
   System.String strNameSpace,
   int iKPP,
   boolean bUseSecureAuth
);
public System.String AuthUrl(
   System.String strReturnUrl,
   int iTimeWindow,
   int iForceLogin,
   System.String strCoBrandedArgs,
   int iLangID,
   System.String strNameSpace,
   int iKPP,
   int iUseSecureAuth
);
public System.String AuthUrl2();
public System.String AuthUrl2(
   System.String strReturnUrl,
   int iTimeWindow,
   boolean fForceLogin,
```

```
   System.String strCoBrandedArgs,
   int iLangID,
   System.String strNameSpace,
   int iKPP,
   boolean bUseSecureAuth
);
public System.String AuthUrl2(
   System.String strReturnUrl,
   int iTimeWindow,
   int iForceLogin,
   System.String strCoBrandedArgs,
   int iLangID,
   System.String strNameSpace,
   int iKPP,
   int iUseSecureAuth
);
public static System.String Compress(
   System.String strData
);
public static boolean CryptIsValid();
public static int CryptPutHost(
   System.String strHost
);
public static int CryptPutSite(
   System.String strSite
);
public static System.String Decompress(
   System.String strData
);
public static System.String Decrypt(
   System.String strData
);
public static System.String Encrypt(
   System.String strData
);
public System.Object GetCurrentConfig(
   System.String strAttribute
);
public System.String GetDomainAttribute(
   System.String strAttribute,
   int iLCID,
   System.String strDomain
);
public System.String GetDomainFromMemberName(
   System.String strMemberName
);
public boolean GetIsAuthenticated(
   int iTimeWindow,
   boolean bForceLogin,
   boolean bCheckSecure
);
public boolean GetIsAuthenticated(
   int iTimeWindow,
   int iForceLogin,
   int iCheckSecure
);
public System.String GetLoginChallenge();
public System.String GetLoginChallenge(
   System.String szRetURL,
   int iTimeWindow,
   int fForceLogin,
   System.String szCOBrandArgs,
   int iLangID,
   System.String strNameSpace,
   int iKPP,
   int iUseSecureAuth,
   System.Object oExtraParams
);
```

```
public System.Object GetOption(
   System.String strOpt
);
public System.Object GetProfileObject(
   System.String strProfileName
);
public boolean HasFlag(
   int iFlagMask
);
public boolean HasProfile(
   System.String strProfile
);
public boolean HaveConsent(
   boolean bNeedFullConsent,
   boolean bNeedBirthdate
);
public int LoginUser();
public int LoginUser(
   System.String szRetURL,
   int iTimeWindow,
   boolean fForceLogin,
   System.String szCOBrandArgs,
   int iLangID,
   System.String strNameSpace,
   int iKPP,
   boolean fUseSecureAuth,
   System.Object oExtraParams
);
public int LoginUser(
   System.String szRetURL,
   int iTimeWindow,
   int fForceLogin,
   System.String szCOBrandArgs,
   int iLangID,
   System.String strNameSpace,
   int iKPP,
   int iUseSecureAuth,
   System.Object oExtraParams
);
public System.String LogoTag();
public System.String LogoTag(
   System.String strReturnUrl,
   int iTimeWindow,
   boolean fForceLogin,
   System.String strCoBrandedArgs,
   int iLangID,
   boolean fSecure,
   System.String strNameSpace,
   int iKPP,
   boolean bUseSecureAuth
);
public System.String LogoTag(
   System.String strReturnUrl,
   int iTimeWindow,
   int iForceLogin,
   System.String strCoBrandedArgs,
   int iLangID,
   int iSecure,
   System.String strNameSpace,
   int iKPP,
   int iUseSecureAuth
);
public System.String LogoTag2();
public System.String LogoTag2(
   System.String strReturnUrl,
   int iTimeWindow,
   boolean fForceLogin,
   System.String strCoBrandedArgs,
```

```
      int iLangID,
      boolean fSecure,
      System.String strNameSpace,
      int iKPP,
      boolean bUseSecureAuth
   );
   public System.String LogoTag2(
      System.String strReturnUrl,
      int iTimeWindow,
      int iForceLogin,
      System.String strCoBrandedArgs,
      int iLangID,
      int iSecure,
      System.String strNameSpace,
      int iKPP,
      int iUseSecureAuth
   );
   public System.String LogoutURL();
   public System.String LogoutURL(
      System.String szReturnURL,
      System.String szCOBrandArgs,
      int iLangID,
      System.String strDomain,
      int iUseSecureAuth
   );
   public void SetOption(
      System.String strOpt,
      System.Object vOpt
   );
   public static void SignOut(
      System.String strSignOutDotGifFileName
   );
   public System.Object Ticket(
      System.String strAttribute
   );
```

Protected Methods

```
   protected void Finalize();
```

UrlAuthorizationModule Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UrlAuthorizationModule
   extends System.Object
   implements System.Web.IHttpModule
```

Public Constructors

```
public UrlAuthorizationModule();
```

Public Methods

```
public final void Dispose();
public final void Init(
   System.Web.HttpApplication app
);
```

WindowsAuthenticationEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WindowsAuthenticationEventArgs
   extends System.EventArgs
```

Public Constructors

```
public WindowsAuthenticationEventArgs(
   System.Security.Principal.WindowsIdentity identity,
   System.Web.HttpContext context
);
```

Public Properties

```
public System.Web.HttpContext get_Context();
public System.Security.Principal.WindowsIdentity get_Identity();
public System.Security.Principal.IPrincipal get_User();
public void set_User(
   System.Security.Principal.IPrincipal value
);
```

WindowsAuthenticationEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void WindowsAuthenticationEventHandler(
   System.Object sender,
   System.Web.Security.WindowsAuthenticationEventArgs e
);
```

WindowsAuthenticationModule Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WindowsAuthenticationModule
   extends System.Object
   implements System.Web.IHttpModule
```

Public Constructors

```
public WindowsAuthenticationModule();
```

Public Methods

```
public final void Dispose();
public final void Init(
   System.Web.HttpApplication app
);
```

Public Events

```
public void add_Authenticate(
   System.Web.Security.WindowsAuthenticationEventHandler value
);
public void remove_Authenticate(
   System.Web.Security.WindowsAuthenticationEventHandler value
);
```

System.Web.Services Namespace (Visual J# Syntax)

Classes

WebMethodAttribute

WebService

WebServiceAttribute

WebServiceBindingAttribute

WebMethodAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = true) */
public class WebMethodAttribute
   extends System.Attribute
```

Public Constructors

```
public WebMethodAttribute();
public WebMethodAttribute(
   boolean enableSession
);
public WebMethodAttribute(
   boolean enableSession,
   System.EnterpriseServices.TransactionOption transactionOption
);
public WebMethodAttribute(
   boolean enableSession,
   System.EnterpriseServices.TransactionOption transactionOption,
   int cacheDuration
);
public WebMethodAttribute(
   boolean enableSession,
   System.EnterpriseServices.TransactionOption transactionOption,
   int cacheDuration,
   boolean bufferResponse
);
```

Public Properties

```
public boolean get_BufferResponse();
public void set_BufferResponse(
   boolean value
);
public int get_CacheDuration();
public void set_CacheDuration(
   int value
);
public System.String get_Description();
public void set_Description(
   System.String value
);
public boolean get_EnableSession();
public void set_EnableSession(
   boolean value
);
public System.String get_MessageName();
public void set_MessageName(
   System.String value
);
public System.EnterpriseServices.TransactionOption get_TransactionOption();
public void set_TransactionOption(
   System.EnterpriseServices.TransactionOption value
);
```

WebService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WebService
   extends System.ComponentModel.MarshalByValueComponent
```

Public Constructors

```
public WebService();
```

Public Properties

```
public System.Web.HttpApplicationState get_Application();
public System.Web.HttpContext get_Context();
public System.Web.HttpServerUtility get_Server();
public System.Web.SessionState.HttpSessionState get_Session();
public System.Security.Principal.IPrincipal get_User();
```

WebServiceAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class WebServiceAttribute
   extends System.Attribute
```

Public Constructors

```
public WebServiceAttribute();
```

Public Properties

```
public System.String get_Description();
public void set_Description(
   System.String value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
```

Public Fields

```
public static System.String DefaultNamespace;
```

WebServiceBindingAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = true, Inherite
d = true) */
public class WebServiceBindingAttribute
   extends System.Attribute
```

Public Constructors

```
public WebServiceBindingAttribute();
public WebServiceBindingAttribute(
   System.String name
);
public WebServiceBindingAttribute(
   System.String name,
   System.String ns
);
public WebServiceBindingAttribute(
   System.String name,
   System.String ns,
   System.String location
);
```

Public Properties

```
public System.String get_Location();
public void set_Location(
   System.String value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
```

System.Web.Services.Configuration Namespace (Visual J# Syntax)

Classes

XmlFormatExtensionAttribute

XmlFormatExtensionPointAttribute

XmlFormatExtensionPrefixAttribute

XmlFormatExtensionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class XmlFormatExtensionAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlFormatExtensionAttribute();
public XmlFormatExtensionAttribute(
   System.String elementName,
   System.String ns,
   System.Type extensionPoint1
);
public XmlFormatExtensionAttribute(
   System.String elementName,
   System.String ns,
   System.Type[] extensionPoints
);
public XmlFormatExtensionAttribute(
   System.String elementName,
   System.String ns,
   System.Type extensionPoint1,
   System.Type extensionPoint2
);
public XmlFormatExtensionAttribute(
   System.String elementName,
   System.String ns,
   System.Type extensionPoint1,
   System.Type extensionPoint2,
   System.Type extensionPoint3
);
public XmlFormatExtensionAttribute(
   System.String elementName,
   System.String ns,
   System.Type extensionPoint1,
   System.Type extensionPoint2,
   System.Type extensionPoint3,
   System.Type extensionPoint4
);
```

Public Properties

```
public System.String get_ElementName();
public void set_ElementName(
   System.String value
);
public System.Type[] get_ExtensionPoints();
public void set_ExtensionPoints(
   System.Type[] value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
```

XmlFormatExtensionPointAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class XmlFormatExtensionPointAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlFormatExtensionPointAttribute(
   System.String memberName
);
```

Public Properties

```
public boolean get_AllowElements();
public void set_AllowElements(
   boolean value
);
public System.String get_MemberName();
public void set_MemberName(
   System.String value
);
```

XmlFormatExtensionPrefixAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = true, Inherite
d = true) */
public class XmlFormatExtensionPrefixAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlFormatExtensionPrefixAttribute();
public XmlFormatExtensionPrefixAttribute(
   System.String prefix,
   System.String ns
);
```

Public Properties

```
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.String get_Prefix();
public void set_Prefix(
   System.String value
);
```

System.Web.Services.Description Namespace (Visual J# Syntax)

Classes

Binding

BindingCollection

DocumentableItem

FaultBinding

FaultBindingCollection

HttpAddressBinding

HttpBinding

HttpOperationBinding

HttpUrlEncodedBinding

HttpUrlReplacementBinding

Import

ImportCollection

InputBinding

Message

MessageBinding

MessageCollection

MessagePart

MessagePartCollection

MimeContentBinding

MimeMultipartRelatedBinding

MimePart

MimePartCollection

MimeTextBinding

MimeTextMatch

MimeTextMatchCollection

MimeXmlBinding

Operation

OperationBinding

OperationBindingCollection

OperationCollection

OperationFault

OperationFaultCollection

OperationInput

OperationMessage

OperationMessageCollection

OperationOutput

OutputBinding

Port

PortCollection

PortType

PortTypeCollection

ProtocolImporter

ProtocolReflector

Service

ServiceCollection

ServiceDescription

ServiceDescriptionBaseCollection

ServiceDescriptionCollection

ServiceDescriptionFormatExtension

ServiceDescriptionFormatExtensionCollection

ServiceDescriptionImporter

ServiceDescriptionReflector

SoapAddressBinding

SoapBinding

SoapBodyBinding

SoapExtensionImporter

SoapExtensionReflector

SoapFaultBinding

SoapHeaderBinding

SoapHeaderFaultBinding

SoapOperationBinding

SoapProtocolImporter

SoapTransportImporter

Types

Enumerations

OperationFlow

ServiceDescriptionImportStyle

ServiceDescriptionImportWarnings

SoapBindingStyle

SoapBindingUse

Binding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionPointAttribute(MemberName
= "Extensions", AllowElements = true) */
public class Binding
   extends System.Web.Services.Description.DocumentableItem
```

Public Constructors

```
public Binding();
```

Public Properties

```
public System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection get_Extens
ions();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Web.Services.Description.OperationBindingCollection get_Operations();
public System.Web.Services.Description.ServiceDescription get_ServiceDescription();
public System.Xml.XmlQualifiedName get_Type();
public void set_Type(
   System.Xml.XmlQualifiedName value
);
```

BindingCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class BindingCollection
   extends System.Web.Services.Description.ServiceDescriptionBaseCollection
```

Public Properties

```
public System.Web.Services.Description.Binding get_Item(
   int index
);
public System.Web.Services.Description.Binding get_Item(
   System.String name
);
public void set_Item(
   int index,
   System.Web.Services.Description.Binding value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Description.Binding binding
);
public boolean Contains(
   System.Web.Services.Description.Binding binding
);
public void CopyTo(
   System.Web.Services.Description.Binding[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Description.Binding binding
);
public void Insert(
   int index,
   System.Web.Services.Description.Binding binding
);
public void Remove(
   System.Web.Services.Description.Binding binding
);
```

Protected Methods

```
protected System.String GetKey(
   System.Object value
);
protected void SetParent(
   System.Object value,
   System.Object parent
);
```

DocumentableItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class DocumentableItem
   extends System.Object
```

Public Properties

```
public System.String get_Documentation();
public void set_Documentation(
   System.String value
);
```

Protected Constructors

```
protected DocumentableItem();
```

FaultBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionPointAttribute(MemberName
= "Extensions", AllowElements = true) */
public class FaultBinding
   extends System.Web.Services.Description.MessageBinding
```

Public Constructors

```
public FaultBinding();
```

Public Properties

```
public System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection get_Extens
ions();
```

FaultBindingCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class FaultBindingCollection
   extends System.Web.Services.Description.ServiceDescriptionBaseCollection
```

Public Properties

```
public System.Web.Services.Description.FaultBinding get_Item(
   int index
);
public System.Web.Services.Description.FaultBinding get_Item(
   System.String name
);
public void set_Item(
   int index,
   System.Web.Services.Description.FaultBinding value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Description.FaultBinding bindingOperationFault
);
public boolean Contains(
   System.Web.Services.Description.FaultBinding bindingOperationFault
);
public void CopyTo(
   System.Web.Services.Description.FaultBinding[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Description.FaultBinding bindingOperationFault
);
public void Insert(
   int index,
   System.Web.Services.Description.FaultBinding bindingOperationFault
);
public void Remove(
   System.Web.Services.Description.FaultBinding bindingOperationFault
);
```

Protected Methods

```
protected System.String GetKey(
   System.Object value
);
protected void SetParent(
   System.Object value,
   System.Object parent
);
```

HttpAddressBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/http/", ElementName = "address"
) */
public class HttpAddressBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public HttpAddressBinding();
```

Public Properties

```
public System.String get_Location();
public void set_Location(
   System.String value
);
```

HttpBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/http/", ElementName = "binding"
) */
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionPrefixAttribute(Prefix = "
http", Namespace = "http://schemas.xmlsoap.org/wsdl/http/") */
public class HttpBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public HttpBinding();
```

Public Properties

```
public System.String get_Verb();
public void set_Verb(
   System.String value
);
```

Public Fields

```
public static System.String Namespace;
```

HttpOperationBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/http/", ElementName = "operatio
n") */
public class HttpOperationBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public HttpOperationBinding();
```

Public Properties

```
public System.String get_Location();
public void set_Location(
   System.String value
);
```

HttpUrlEncodedBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/http/", ElementName = "urlEncod
ed") */
public class HttpUrlEncodedBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public HttpUrlEncodedBinding();
```

HttpUrlReplacementBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/http/", ElementName = "urlRepla
cement") */
public class HttpUrlReplacementBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public HttpUrlReplacementBinding();
```

Import Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Import
   extends System.Web.Services.Description.DocumentableItem
```

Public Constructors

```
public Import();
```

Public Properties

```
public System.String get_Location();
public void set_Location(
   System.String value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.Web.Services.Description.ServiceDescription get_ServiceDescription();
```

ImportCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ImportCollection
   extends System.Web.Services.Description.ServiceDescriptionBaseCollection
```

Public Properties

```
public System.Web.Services.Description.Import get_Item(
   int index
);
public void set_Item(
   int index,
   System.Web.Services.Description.Import value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Description.Import import
);
public boolean Contains(
   System.Web.Services.Description.Import import
);
public void CopyTo(
   System.Web.Services.Description.Import[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Description.Import import
);
public void Insert(
   int index,
   System.Web.Services.Description.Import import
);
public void Remove(
   System.Web.Services.Description.Import import
);
```

Protected Methods

```
protected void SetParent(
   System.Object value,
   System.Object parent
);
```

InputBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionPointAttribute(MemberName
= "Extensions", AllowElements = true) */
public class InputBinding
   extends System.Web.Services.Description.MessageBinding
```

Public Constructors

```
public InputBinding();
```

Public Properties

```
public System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection get_Extens
ions();
```

Message Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Message
   extends System.Web.Services.Description.DocumentableItem
```

Public Constructors

```
public Message();
```

Public Properties

```
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Web.Services.Description.MessagePartCollection get_Parts();
public System.Web.Services.Description.ServiceDescription get_ServiceDescription();
```

Public Methods

```
public System.Web.Services.Description.MessagePart FindPartByName(
   System.String partName
);
public System.Web.Services.Description.MessagePart[] FindPartsByName(
   System.String[] partNames
);
```

MessageBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class MessageBinding
   extends System.Web.Services.Description.DocumentableItem
```

Public Properties

```
public abstract System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection g
et_Extensions();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Web.Services.Description.OperationBinding get_OperationBinding();
```

Protected Constructors

```
protected MessageBinding();
```

MessageCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class MessageCollection
   extends System.Web.Services.Description.ServiceDescriptionBaseCollection
```

Public Properties

```
public System.Web.Services.Description.Message get_Item(
   int index
);
public System.Web.Services.Description.Message get_Item(
   System.String name
);
public void set_Item(
   int index,
   System.Web.Services.Description.Message value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Description.Message message
);
public boolean Contains(
   System.Web.Services.Description.Message message
);
public void CopyTo(
   System.Web.Services.Description.Message[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Description.Message message
);
public void Insert(
   int index,
   System.Web.Services.Description.Message message
);
public void Remove(
   System.Web.Services.Description.Message message
);
```

Protected Methods

```
protected System.String GetKey(
   System.Object value
);
protected void SetParent(
   System.Object value,
   System.Object parent
);
```

MessagePart Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MessagePart
   extends System.Web.Services.Description.DocumentableItem
```

Public Constructors

```
public MessagePart();
```

Public Properties

```
public System.Xml.XmlQualifiedName get_Element();
public void set_Element(
   System.Xml.XmlQualifiedName value
);
public System.Web.Services.Description.Message get_Message();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Xml.XmlQualifiedName get_Type();
public void set_Type(
   System.Xml.XmlQualifiedName value
);
```

MessagePartCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class MessagePartCollection
   extends System.Web.Services.Description.ServiceDescriptionBaseCollection
```

Public Properties

```
public System.Web.Services.Description.MessagePart get_Item(
   int index
);
public System.Web.Services.Description.MessagePart get_Item(
   System.String name
);
public void set_Item(
   int index,
   System.Web.Services.Description.MessagePart value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Description.MessagePart messagePart
);
public boolean Contains(
   System.Web.Services.Description.MessagePart messagePart
);
public void CopyTo(
   System.Web.Services.Description.MessagePart[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Description.MessagePart messagePart
);
public void Insert(
   int index,
   System.Web.Services.Description.MessagePart messagePart
);
public void Remove(
   System.Web.Services.Description.MessagePart messagePart
);
```

Protected Methods

```
protected System.String GetKey(
   System.Object value
);
protected void SetParent(
   System.Object value,
   System.Object parent
);
```

MimeContentBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionPrefixAttribute(Prefix = "
mime", Namespace = "http://schemas.xmlsoap.org/wsdl/mime/") */
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/mime/", ElementName = "content"
) */
public class MimeContentBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public MimeContentBinding();
```

Public Properties

```
public System.String get_Part();
public void set_Part(
   System.String value
);
public System.String get_Type();
public void set_Type(
   System.String value
);
```

Public Fields

```
public static System.String Namespace;
```

MimeMultipartRelatedBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/mime/", ElementName = "multipar
tRelated") */
public class MimeMultipartRelatedBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public MimeMultipartRelatedBinding();
```

Public Properties

```
public System.Web.Services.Description.MimePartCollection get_Parts();
```

MimePart Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionPointAttribute(MemberName
= "Extensions", AllowElements = true) */
public class MimePart
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public MimePart();
```

Public Properties

```
public System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection get_Extens
ions();
```

MimePartCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class MimePartCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public MimePartCollection();
```

Public Properties

```
public System.Web.Services.Description.MimePart get_Item(
   int index
);
public void set_Item(
   int index,
   System.Web.Services.Description.MimePart value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Description.MimePart mimePart
);
public boolean Contains(
   System.Web.Services.Description.MimePart mimePart
);
public void CopyTo(
   System.Web.Services.Description.MimePart[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Description.MimePart mimePart
);
public void Insert(
   int index,
   System.Web.Services.Description.MimePart mimePart
);
public void Remove(
   System.Web.Services.Description.MimePart mimePart
);
```

MimeTextBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://microsoft.com/wsdl/mime/textMatching/", ElementName = "t
ext") */
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionPrefixAttribute(Prefix = "
tm", Namespace = "http://microsoft.com/wsdl/mime/textMatching/") */
public class MimeTextBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public MimeTextBinding();
```

Public Properties

```
public System.Web.Services.Description.MimeTextMatchCollection get_Matches();
```

Public Fields

```
public static System.String Namespace;
```

MimeTextMatch Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MimeTextMatch
   extends System.Object
```

Public Constructors

```
public MimeTextMatch();
```

Public Properties

```
public int get_Capture();
public void set_Capture(
   int value
);
public int get_Group();
public void set_Group(
   int value
);
public boolean get_IgnoreCase();
public void set_IgnoreCase(
   boolean value
);
public System.Web.Services.Description.MimeTextMatchCollection get_Matches();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_Pattern();
public void set_Pattern(
   System.String value
);
public int get_Repeats();
public void set_Repeats(
   int value
);
public System.String get_RepeatsString();
public void set_RepeatsString(
   System.String value
);
public System.String get_Type();
public void set_Type(
   System.String value
);
```

MimeTextMatchCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class MimeTextMatchCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public MimeTextMatchCollection();
```

Public Properties

```
public System.Web.Services.Description.MimeTextMatch get_Item(
   int index
);
public void set_Item(
   int index,
   System.Web.Services.Description.MimeTextMatch value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Description.MimeTextMatch match
);
public boolean Contains(
   System.Web.Services.Description.MimeTextMatch match
);
public void CopyTo(
   System.Web.Services.Description.MimeTextMatch[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Description.MimeTextMatch match
);
public void Insert(
   int index,
   System.Web.Services.Description.MimeTextMatch match
);
public void Remove(
   System.Web.Services.Description.MimeTextMatch match
);
```

MimeXmlBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/mime/", ElementName = "mimeXml"
) */
public class MimeXmlBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public MimeXmlBinding();
```

Public Properties

```
public System.String get_Part();
public void set_Part(
   System.String value
);
```

Operation Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Operation
   extends System.Web.Services.Description.DocumentableItem
```

Public Constructors

```
public Operation();
```

Public Properties

```
public System.Web.Services.Description.OperationFaultCollection get_Faults();
public System.Web.Services.Description.OperationMessageCollection get_Messages();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String[] get_ParameterOrder();
public void set_ParameterOrder(
   System.String[] value
);
public System.String get_ParameterOrderString();
public void set_ParameterOrderString(
   System.String value
);
public System.Web.Services.Description.PortType get_PortType();
```

Public Methods

```
public boolean IsBoundBy(
   System.Web.Services.Description.OperationBinding operationBinding
);
```

OperationBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionPointAttribute(MemberName
= "Extensions", AllowElements = true) */
public class OperationBinding
   extends System.Web.Services.Description.DocumentableItem
```

Public Constructors

```
public OperationBinding();
```

Public Properties

```
public System.Web.Services.Description.Binding get_Binding();
public System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection get_Extens
ions();
public System.Web.Services.Description.FaultBindingCollection get_Faults();
public System.Web.Services.Description.InputBinding get_Input();
public void set_Input(
   System.Web.Services.Description.InputBinding value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Web.Services.Description.OutputBinding get_Output();
public void set_Output(
   System.Web.Services.Description.OutputBinding value
);
```

OperationBindingCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class OperationBindingCollection
   extends System.Web.Services.Description.ServiceDescriptionBaseCollection
```

Public Properties

```
public System.Web.Services.Description.OperationBinding get_Item(
   int index
);
public void set_Item(
   int index,
   System.Web.Services.Description.OperationBinding value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Description.OperationBinding bindingOperation
);
public boolean Contains(
   System.Web.Services.Description.OperationBinding bindingOperation
);
public void CopyTo(
   System.Web.Services.Description.OperationBinding[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Description.OperationBinding bindingOperation
);
public void Insert(
   int index,
   System.Web.Services.Description.OperationBinding bindingOperation
);
public void Remove(
   System.Web.Services.Description.OperationBinding bindingOperation
);
```

Protected Methods

```
protected void SetParent(
   System.Object value,
   System.Object parent
);
```

OperationCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class OperationCollection
   extends System.Web.Services.Description.ServiceDescriptionBaseCollection
```

Public Properties

```
public System.Web.Services.Description.Operation get_Item(
   int index
);
public void set_Item(
   int index,
   System.Web.Services.Description.Operation value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Description.Operation operation
);
public boolean Contains(
   System.Web.Services.Description.Operation operation
);
public void CopyTo(
   System.Web.Services.Description.Operation[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Description.Operation operation
);
public void Insert(
   int index,
   System.Web.Services.Description.Operation operation
);
public void Remove(
   System.Web.Services.Description.Operation operation
);
```

Protected Methods

```
protected void SetParent(
   System.Object value,
   System.Object parent
);
```

OperationFault Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class OperationFault
   extends System.Web.Services.Description.OperationMessage
```

Public Constructors

```
public OperationFault();
```

OperationFaultCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class OperationFaultCollection
   extends System.Web.Services.Description.ServiceDescriptionBaseCollection
```

Public Properties

```
public System.Web.Services.Description.OperationFault get_Item(
   int index
);
public System.Web.Services.Description.OperationFault get_Item(
   System.String name
);
public void set_Item(
   int index,
   System.Web.Services.Description.OperationFault value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Description.OperationFault operationFaultMessage
);
public boolean Contains(
   System.Web.Services.Description.OperationFault operationFaultMessage
);
public void CopyTo(
   System.Web.Services.Description.OperationFault[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Description.OperationFault operationFaultMessage
);
public void Insert(
   int index,
   System.Web.Services.Description.OperationFault operationFaultMessage
);
public void Remove(
   System.Web.Services.Description.OperationFault operationFaultMessage
);
```

Protected Methods

```
protected System.String GetKey(
   System.Object value
);
protected void SetParent(
   System.Object value,
   System.Object parent
);
```

OperationFlow Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum OperationFlow
```

Members

None

Notification

OneWay

RequestResponse

SolicitResponse

OperationInput Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class OperationInput
   extends System.Web.Services.Description.OperationMessage
```

Public Constructors

```
public OperationInput();
```

OperationMessage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class OperationMessage
   extends System.Web.Services.Description.DocumentableItem
```

Public Properties

```
public System.Xml.XmlQualifiedName get_Message();
public void set_Message(
   System.Xml.XmlQualifiedName value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Web.Services.Description.Operation get_Operation();
```

Protected Constructors

```
protected OperationMessage();
```

OperationMessageCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class OperationMessageCollection
   extends System.Web.Services.Description.ServiceDescriptionBaseCollection
```

Public Properties

```
public System.Web.Services.Description.OperationFlow get_Flow();
public System.Web.Services.Description.OperationInput get_Input();
public System.Web.Services.Description.OperationMessage get_Item(
   int index
);
public void set_Item(
   int index,
   System.Web.Services.Description.OperationMessage value
);
public System.Web.Services.Description.OperationOutput get_Output();
```

Public Methods

```
public int Add(
   System.Web.Services.Description.OperationMessage operationMessage
);
public boolean Contains(
   System.Web.Services.Description.OperationMessage operationMessage
);
public void CopyTo(
   System.Web.Services.Description.OperationMessage[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Description.OperationMessage operationMessage
);
public void Insert(
   int index,
   System.Web.Services.Description.OperationMessage operationMessage
);
public void Remove(
   System.Web.Services.Description.OperationMessage operationMessage
);
```

Protected Methods

```
protected void OnInsert(
   int index,
   System.Object value
);
protected void OnSet(
   int index,
   System.Object oldValue,
   System.Object newValue
);
protected void OnValidate(
   System.Object value
```

```
);
protected void SetParent(
   System.Object value,
   System.Object parent
);
```

OperationOutput Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class OperationOutput
   extends System.Web.Services.Description.OperationMessage
```

Public Constructors

```
public OperationOutput();
```

OutputBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionPointAttribute(MemberName
= "Extensions", AllowElements = true) */
public class OutputBinding
   extends System.Web.Services.Description.MessageBinding
```

Public Constructors

```
public OutputBinding();
```

Public Properties

```
public System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection get_Extens
ions();
```

Port Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionPointAttribute(MemberName
= "Extensions", AllowElements = true) */
public class Port
   extends System.Web.Services.Description.DocumentableItem
```

Public Constructors

```
public Port();
```

Public Properties

```
public System.Xml.XmlQualifiedName get_Binding();
public void set_Binding(
   System.Xml.XmlQualifiedName value
);
public System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection get_Extens
ions();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Web.Services.Description.Service get_Service();
```

PortCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class PortCollection
   extends System.Web.Services.Description.ServiceDescriptionBaseCollection
```

Public Properties

```
public System.Web.Services.Description.Port get_Item(
   int index
);
public System.Web.Services.Description.Port get_Item(
   System.String name
);
public void set_Item(
   int index,
   System.Web.Services.Description.Port value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Description.Port port
);
public boolean Contains(
   System.Web.Services.Description.Port port
);
public void CopyTo(
   System.Web.Services.Description.Port[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Description.Port port
);
public void Insert(
   int index,
   System.Web.Services.Description.Port port
);
public void Remove(
   System.Web.Services.Description.Port port
);
```

Protected Methods

```
protected System.String GetKey(
   System.Object value
);
protected void SetParent(
   System.Object value,
   System.Object parent
);
```

PortType Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PortType
   extends System.Web.Services.Description.DocumentableItem
```

Public Constructors

```
public PortType();
```

Public Properties

```
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Web.Services.Description.OperationCollection get_Operations();
public System.Web.Services.Description.ServiceDescription get_ServiceDescription();
```

PortTypeCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class PortTypeCollection
   extends System.Web.Services.Description.ServiceDescriptionBaseCollection
```

Public Properties

```
public System.Web.Services.Description.PortType get_Item(
   int index
);
public System.Web.Services.Description.PortType get_Item(
   System.String name
);
public void set_Item(
   int index,
   System.Web.Services.Description.PortType value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Description.PortType portType
);
public boolean Contains(
   System.Web.Services.Description.PortType portType
);
public void CopyTo(
   System.Web.Services.Description.PortType[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Description.PortType portType
);
public void Insert(
   int index,
   System.Web.Services.Description.PortType portType
);
public void Remove(
   System.Web.Services.Description.PortType portType
);
```

Protected Methods

```
protected System.String GetKey(
   System.Object value
);
protected void SetParent(
   System.Object value,
   System.Object parent
);
```

ProtocolImporter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class ProtocolImporter
   extends System.Object
```

Public Properties

```
public System.Xml.Serialization.XmlSchemas get_AbstractSchemas();
public System.Web.Services.Description.Binding get_Binding();
public System.String get_ClassName();
public System.Xml.Serialization.CodeIdentifiers get_ClassNames();
public System.CodeDom.CodeNamespace get_CodeNamespace();
public System.CodeDom.CodeTypeDeclaration get_CodeTypeDeclaration();
public System.Xml.Serialization.XmlSchemas get_ConcreteSchemas();
public System.Web.Services.Description.Message get_InputMessage();
public System.String get_MethodName();
public System.Web.Services.Description.Operation get_Operation();
public System.Web.Services.Description.OperationBinding get_OperationBinding();
public System.Web.Services.Description.Message get_OutputMessage();
public System.Web.Services.Description.Port get_Port();
public System.Web.Services.Description.PortType get_PortType();
public abstract System.String get_ProtocolName();
public System.Xml.Serialization.XmlSchemas get_Schemas();
public System.Web.Services.Description.Service get_Service();
public System.Web.Services.Description.ServiceDescriptionCollection get_ServiceDescriptions()
;
public System.Web.Services.Description.ServiceDescriptionImportStyle get_Style();
public System.Web.Services.Description.ServiceDescriptionImportWarnings get_Warnings();
public void set_Warnings(
   System.Web.Services.Description.ServiceDescriptionImportWarnings value
);
```

Public Methods

```
public void AddExtensionWarningComments(
   System.CodeDom.CodeCommentStatementCollection comments,
   System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection extensions
);
public System.Exception OperationBindingSyntaxException(
   System.String text
);
public System.Exception OperationSyntaxException(
   System.String text
);
public void UnsupportedBindingWarning(
   System.String text
);
public void UnsupportedOperationBindingWarning(
   System.String text
);
public void UnsupportedOperationWarning(
   System.String text
);
```

Protected Constructors

```
protected ProtocolImporter();
```

Protected Methods

```
protected abstract System.CodeDom.CodeTypeDeclaration BeginClass();
protected void BeginNamespace();
protected void EndClass();
protected void EndNamespace();
protected abstract System.CodeDom.CodeMemberMethod GenerateMethod();
protected abstract boolean IsBindingSupported();
protected abstract boolean IsOperationFlowSupported(
   System.Web.Services.Description.OperationFlow flow
);
```

ProtocolReflector Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class ProtocolReflector
   extends System.Object
```

Public Properties

```
public System.Web.Services.Description.Binding get_Binding();
public System.String get_DefaultNamespace();
public System.Web.Services.Description.MessageCollection get_HeaderMessages();
public System.Web.Services.Description.Message get_InputMessage();
public System.Web.Services.Protocols.LogicalMethodInfo get_Method();
public System.Web.Services.WebMethodAttribute get_MethodAttribute();
public System.Web.Services.Protocols.LogicalMethodInfo[] get_Methods();
public System.Web.Services.Description.Operation get_Operation();
public System.Web.Services.Description.OperationBinding get_OperationBinding();
public System.Web.Services.Description.Message get_OutputMessage();
public System.Web.Services.Description.Port get_Port();
public System.Web.Services.Description.PortType get_PortType();
public abstract System.String get_ProtocolName();
public System.Xml.Serialization.XmlReflectionImporter get_ReflectionImporter();
public System.Xml.Serialization.XmlSchemaExporter get_SchemaExporter();
public System.Xml.Serialization.XmlSchemas get_Schemas();
public System.Web.Services.Description.Service get_Service();
public System.Web.Services.Description.ServiceDescription get_ServiceDescription();
public System.Web.Services.Description.ServiceDescriptionCollection get_ServiceDescriptions()
;
public System.Type get_ServiceType();
public System.String get_ServiceUrl();
```

Public Methods

```
public System.Web.Services.Description.ServiceDescription GetServiceDescription(
   System.String ns
);
```

Protected Constructors

```
protected ProtocolReflector();
```

Protected Methods

```
protected void BeginClass();
protected void EndClass();
protected abstract boolean ReflectMethod();
protected System.String ReflectMethodBinding();
```

Service Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Service
   extends System.Web.Services.Description.DocumentableItem
```

Public Constructors

```
public Service();
```

Public Properties

```
public System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection get_Extens
ions();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Web.Services.Description.PortCollection get_Ports();
public System.Web.Services.Description.ServiceDescription get_ServiceDescription();
```

ServiceCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ServiceCollection
   extends System.Web.Services.Description.ServiceDescriptionBaseCollection
```

Public Properties

```
public System.Web.Services.Description.Service get_Item(
   int index
);
public System.Web.Services.Description.Service get_Item(
   System.String name
);
public void set_Item(
   int index,
   System.Web.Services.Description.Service value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Description.Service service
);
public boolean Contains(
   System.Web.Services.Description.Service service
);
public void CopyTo(
   System.Web.Services.Description.Service[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Description.Service service
);
public void Insert(
   int index,
   System.Web.Services.Description.Service service
);
public void Remove(
   System.Web.Services.Description.Service service
);
```

Protected Methods

```
protected System.String GetKey(
   System.Object value
);
protected void SetParent(
   System.Object value,
   System.Object parent
);
```

ServiceDescription Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionPointAttribute(MemberName
= "Extensions", AllowElements = true) */
/** @attribute System.Xml.Serialization.XmlRootAttribute(ElementName = "definitions", Namespa
ce = "http://schemas.xmlsoap.org/wsdl/", DataType = "", IsNullable = true) */
public class ServiceDescription
   extends System.Web.Services.Description.DocumentableItem
```

Public Constructors

```
public ServiceDescription();
```

Public Properties

```
public System.Web.Services.Description.BindingCollection get_Bindings();
public System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection get_Extens
ions();
public System.Web.Services.Description.ImportCollection get_Imports();
public System.Web.Services.Description.MessageCollection get_Messages();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Web.Services.Description.PortTypeCollection get_PortTypes();
public System.String get_RetrievalUrl();
public void set_RetrievalUrl(
   System.String value
);
public static System.Xml.Serialization.XmlSerializer get_Serializer();
public System.Web.Services.Description.ServiceDescriptionCollection get_ServiceDescriptions()
;
public System.Web.Services.Description.ServiceCollection get_Services();
public System.String get_TargetNamespace();
public void set_TargetNamespace(
   System.String value
);
public System.Web.Services.Description.Types get_Types();
public void set_Types(
   System.Web.Services.Description.Types value
);
```

Public Methods

```
public static boolean CanRead(
   System.Xml.XmlReader reader
);
public static System.Web.Services.Description.ServiceDescription Read(
   System.IO.Stream stream
);
public static System.Web.Services.Description.ServiceDescription Read(
   System.IO.TextReader textReader
);
public static System.Web.Services.Description.ServiceDescription Read(
   System.String fileName
```

```
);
public static System.Web.Services.Description.ServiceDescription Read(
   System.Xml.XmlReader reader
);
public void Write(
   System.IO.Stream stream
);
public void Write(
   System.IO.TextWriter writer
);
public void Write(
   System.String fileName
);
public void Write(
   System.Xml.XmlWriter writer
);
```

Public Fields

```
public static System.String Namespace;
```

ServiceDescriptionBaseCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class ServiceDescriptionBaseCollection
   extends System.Collections.CollectionBase
```

Protected Properties

```
protected System.Collections.IDictionary get_Table();
```

Protected Methods

```
protected System.String GetKey(
   System.Object value
);
protected void OnClear();
protected void OnInsertComplete(
   int index,
   System.Object value
);
protected void OnRemove(
   int index,
   System.Object value
);
protected void OnSet(
   int index,
   System.Object oldValue,
   System.Object newValue
);
protected void SetParent(
   System.Object value,
   System.Object parent
);
```

ServiceDescriptionCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ServiceDescriptionCollection
   extends System.Web.Services.Description.ServiceDescriptionBaseCollection
```

Public Constructors

```
public ServiceDescriptionCollection();
```

Public Properties

```
public System.Web.Services.Description.ServiceDescription get_Item(
   int index
);
public System.Web.Services.Description.ServiceDescription get_Item(
   System.String ns
);
public void set_Item(
   int index,
   System.Web.Services.Description.ServiceDescription value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Description.ServiceDescription serviceDescription
);
public boolean Contains(
   System.Web.Services.Description.ServiceDescription serviceDescription
);
public void CopyTo(
   System.Web.Services.Description.ServiceDescription[] array,
   int index
);
public System.Web.Services.Description.Binding GetBinding(
   System.Xml.XmlQualifiedName name
);
public System.Web.Services.Description.Message GetMessage(
   System.Xml.XmlQualifiedName name
);
public System.Web.Services.Description.PortType GetPortType(
   System.Xml.XmlQualifiedName name
);
public System.Web.Services.Description.Service GetService(
   System.Xml.XmlQualifiedName name
);
public int IndexOf(
   System.Web.Services.Description.ServiceDescription serviceDescription
);
public void Insert(
   int index,
   System.Web.Services.Description.ServiceDescription serviceDescription
);
public void Remove(
```

```
    System.Web.Services.Description.ServiceDescription serviceDescription
);
```

Protected Methods

```
protected System.String GetKey(
    System.Object value
);
```

ServiceDescriptionFormatExtension Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class ServiceDescriptionFormatExtension
   extends System.Object
```

Public Properties

```
public boolean get_Handled();
public void set_Handled(
   boolean value
);
public System.Object get_Parent();
public boolean get_Required();
public void set_Required(
   boolean value
);
```

Protected Constructors

```
protected ServiceDescriptionFormatExtension();
```

ServiceDescriptionFormatExtensionCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ServiceDescriptionFormatExtensionCollection
   extends System.Web.Services.Description.ServiceDescriptionBaseCollection
```

Public Constructors

```
public ServiceDescriptionFormatExtensionCollection(
   System.Object parent
);
```

Public Properties

```
public System.Object get_Item(
   int index
);
public void set_Item(
   int index,
   System.Object value
);
```

Public Methods

```
public int Add(
   System.Object extension
);
public boolean Contains(
   System.Object extension
);
public void CopyTo(
   System.Object[] array,
   int index
);
public System.Object Find(
   System.Type type
);
public System.Xml.XmlElement Find(
   System.String name,
   System.String ns
);
public System.Object[] FindAll(
   System.Type type
);
public System.Xml.XmlElement[] FindAll(
   System.String name,
   System.String ns
);
public int IndexOf(
   System.Object extension
);
public void Insert(
```

```
   int index,
   System.Object extension
);
public boolean IsHandled(
   System.Object item
);
public boolean IsRequired(
   System.Object item
);
public void Remove(
   System.Object extension
);
```

Protected Methods

```
protected void OnValidate(
   System.Object value
);
protected void SetParent(
   System.Object value,
   System.Object parent
);
```

ServiceDescriptionImporter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ServiceDescriptionImporter
   extends System.Object
```

Public Constructors

```
public ServiceDescriptionImporter();
```

Public Properties

```
public System.String get_ProtocolName();
public void set_ProtocolName(
   System.String value
);
public System.Xml.Serialization.XmlSchemas get_Schemas();
public System.Web.Services.Description.ServiceDescriptionCollection get_ServiceDescriptions()
;
public System.Web.Services.Description.ServiceDescriptionImportStyle get_Style();
public void set_Style(
   System.Web.Services.Description.ServiceDescriptionImportStyle value
);
```

Public Methods

```
public void AddServiceDescription(
   System.Web.Services.Description.ServiceDescription serviceDescription,
   System.String appSettingUrlKey,
   System.String appSettingBaseUrl
);
public System.Web.Services.Description.ServiceDescriptionImportWarnings Import(
   System.CodeDom.CodeNamespace codeNamespace,
   System.CodeDom.CodeCompileUnit codeCompileUnit
);
```

ServiceDescriptionImportStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ServiceDescriptionImportStyle
```

Members

Client

Server

ServiceDescriptionImportWarnings Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ServiceDescriptionImportWarnings
```

Members

NoCodeGenerated

NoMethodsGenerated

OptionalExtensionsIgnored

RequiredExtensionsIgnored

UnsupportedBindingsIgnored

UnsupportedOperationsIgnored

ServiceDescriptionReflector Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ServiceDescriptionReflector
   extends System.Object
```

Public Constructors

```
public ServiceDescriptionReflector();
```

Public Properties

```
public System.Xml.Serialization.XmlSchemas get_Schemas();
public System.Web.Services.Description.ServiceDescriptionCollection get_ServiceDescriptions()
;
```

Public Methods

```
public void Reflect(
   System.Type type,
   System.String url
);
```

SoapAddressBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/soap/", ElementName = "address"
) */
public class SoapAddressBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public SoapAddressBinding();
```

Public Properties

```
public System.String get_Location();
public void set_Location(
   System.String value
);
```

SoapBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionPrefixAttribute(Prefix = "
soap", Namespace = "http://schemas.xmlsoap.org/wsdl/soap/") */
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/soap/", ElementName = "binding"
) */
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionPrefixAttribute(Prefix = "
soapenc", Namespace = "http://schemas.xmlsoap.org/soap/encoding/") */
public class SoapBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public SoapBinding();
```

Public Properties

```
public System.Web.Services.Description.SoapBindingStyle get_Style();
public void set_Style(
   System.Web.Services.Description.SoapBindingStyle value
);
public System.String get_Transport();
public void set_Transport(
   System.String value
);
```

Public Fields

```
public static System.String HttpTransport;
public static System.String Namespace;
```

SoapBindingStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SoapBindingStyle
```

Members

Default

Document

Rpc

SoapBindingUse Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SoapBindingUse
```

Members

Default

Encoded

Literal

SoapBodyBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/soap/", ElementName = "body") *
/
public class SoapBodyBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public SoapBodyBinding();
```

Public Properties

```
public System.String get_Encoding();
public void set_Encoding(
   System.String value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.String[] get_Parts();
public void set_Parts(
   System.String[] value
);
public System.String get_PartsString();
public void set_PartsString(
   System.String value
);
public System.Web.Services.Description.SoapBindingUse get_Use();
public void set_Use(
   System.Web.Services.Description.SoapBindingUse value
);
```

SoapExtensionImporter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class SoapExtensionImporter
   extends System.Object
```

Public Properties

```
public System.Web.Services.Description.SoapProtocolImporter get_ImportContext();
public void set_ImportContext(
   System.Web.Services.Description.SoapProtocolImporter value
);
```

Public Methods

```
public abstract void ImportMethod(
   System.CodeDom.CodeAttributeDeclarationCollection metadata
);
```

Protected Constructors

```
protected SoapExtensionImporter();
```

SoapExtensionReflector Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class SoapExtensionReflector
   extends System.Object
```

Public Properties

```
public System.Web.Services.Description.ProtocolReflector get_ReflectionContext();
public void set_ReflectionContext(
   System.Web.Services.Description.ProtocolReflector value
);
```

Public Methods

```
public abstract void ReflectMethod();
```

Protected Constructors

```
protected SoapExtensionReflector();
```

SoapFaultBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/soap/", ElementName = "fault")
*/
public class SoapFaultBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public SoapFaultBinding();
```

Public Properties

```
public System.String get_Encoding();
public void set_Encoding(
   System.String value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.Web.Services.Description.SoapBindingUse get_Use();
public void set_Use(
   System.Web.Services.Description.SoapBindingUse value
);
```

SoapHeaderBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/soap/", ElementName = "header")
*/
public class SoapHeaderBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public SoapHeaderBinding();
```

Public Properties

```
public System.String get_Encoding();
public void set_Encoding(
   System.String value
);
public boolean get_MapToProperty();
public void set_MapToProperty(
   boolean value
);
public System.Xml.XmlQualifiedName get_Message();
public void set_Message(
   System.Xml.XmlQualifiedName value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.String get_Part();
public void set_Part(
   System.String value
);
public System.Web.Services.Description.SoapBindingUse get_Use();
public void set_Use(
   System.Web.Services.Description.SoapBindingUse value
);
```

SoapHeaderFaultBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/soap/", ElementName = "headerfa
ult") */
public class SoapHeaderFaultBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public SoapHeaderFaultBinding();
```

Public Properties

```
public System.String get_Encoding();
public void set_Encoding(
   System.String value
);
public System.Xml.XmlQualifiedName get_Message();
public void set_Message(
   System.Xml.XmlQualifiedName value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.String get_Part();
public void set_Part(
   System.String value
);
public System.Web.Services.Description.SoapBindingUse get_Use();
public void set_Use(
   System.Web.Services.Description.SoapBindingUse value
);
```

SoapOperationBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionAttribute(ExtensionPoints
= System.Type[], Namespace = "http://schemas.xmlsoap.org/wsdl/soap/", ElementName = "operatio
n") */
public class SoapOperationBinding
   extends System.Web.Services.Description.ServiceDescriptionFormatExtension
```

Public Constructors

```
public SoapOperationBinding();
```

Public Properties

```
public System.String get_SoapAction();
public void set_SoapAction(
   System.String value
);
public System.Web.Services.Description.SoapBindingStyle get_Style();
public void set_Style(
   System.Web.Services.Description.SoapBindingStyle value
);
```

SoapProtocolImporter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapProtocolImporter
   extends System.Web.Services.Description.ProtocolImporter
```

Public Constructors

```
public SoapProtocolImporter();
```

Public Properties

```
public System.String get_ProtocolName();
public System.Web.Services.Description.SoapBinding get_SoapBinding();
public System.Xml.Serialization.SoapCodeExporter get_SoapExporter();
public System.Xml.Serialization.SoapSchemaImporter get_SoapImporter();
public System.Xml.Serialization.XmlCodeExporter get_XmlExporter();
public System.Xml.Serialization.XmlSchemaImporter get_XmlImporter();
```

Protected Methods

```
protected System.CodeDom.CodeTypeDeclaration BeginClass();
protected void BeginNamespace();
protected void EndClass();
protected void EndNamespace();
protected System.CodeDom.CodeMemberMethod GenerateMethod();
protected boolean IsBindingSupported();
protected boolean IsOperationFlowSupported(
   System.Web.Services.Description.OperationFlow flow
);
```

SoapTransportImporter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class SoapTransportImporter
   extends System.Object
```

Public Properties

```
public System.Web.Services.Description.SoapProtocolImporter get_ImportContext();
public void set_ImportContext(
   System.Web.Services.Description.SoapProtocolImporter value
);
```

Public Methods

```
public abstract void ImportClass();
public abstract boolean IsSupportedTransport(
   System.String transport
);
```

Protected Constructors

```
protected SoapTransportImporter();
```

Types Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.Services.Configuration.XmlFormatExtensionPointAttribute(MemberName
= "Extensions", AllowElements = true) */
public class Types
   extends System.Web.Services.Description.DocumentableItem
```

Public Constructors

```
public Types();
```

Public Properties

```
public System.Web.Services.Description.ServiceDescriptionFormatExtensionCollection get_Extens
ions();
public System.Xml.Serialization.XmlSchemas get_Schemas();
```

System.Web.Services.Discovery Namespace (Visual J# Syntax)

Classes

ContractReference

ContractSearchPattern

DiscoveryClientDocumentCollection

DiscoveryClientProtocol

DiscoveryClientProtocol.DiscoveryClientResultsFile

DiscoveryClientReferenceCollection

DiscoveryClientResult

DiscoveryClientResultCollection

DiscoveryDocument

DiscoveryDocumentLinksPattern

DiscoveryDocumentReference

DiscoveryDocumentSearchPattern

DiscoveryExceptionDictionary

DiscoveryReference

DiscoveryReferenceCollection

DiscoveryRequestHandler

DiscoverySearchPattern

DynamicDiscoveryDocument

ExcludePathInfo

SchemaReference

SoapBinding

XmlSchemaSearchPattern

ContractReference Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Xml.Serialization.XmlRootAttribute(ElementName = "contractRef", Namespa
ce = "http://schemas.xmlsoap.org/disco/scl/", DataType = "", IsNullable = true) */
public class ContractReference
   extends System.Web.Services.Discovery.DiscoveryReference
```

Public Constructors

```
public ContractReference();
public ContractReference(
   System.String href
);
public ContractReference(
   System.String href,
   System.String docRef
);
```

Public Properties

```
public System.Web.Services.Description.ServiceDescription get_Contract();
public System.String get_DefaultFilename();
public System.String get_DocRef();
public void set_DocRef(
   System.String value
);
public System.String get_Ref();
public void set_Ref(
   System.String value
);
public System.String get_Url();
public void set_Url(
   System.String value
);
```

Public Methods

```
public System.Object ReadDocument(
   System.IO.Stream stream
);
public void WriteDocument(
   System.Object document,
   System.IO.Stream stream
);
```

Public Fields

```
public static System.String Namespace;
```

Protected Methods

```
protected void Resolve(
   System.String contentType,
   System.IO.Stream stream
);
```

ContractSearchPattern Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ContractSearchPattern
   extends System.Web.Services.Discovery.DiscoverySearchPattern
```

Public Constructors

```
public ContractSearchPattern();
```

Public Properties

```
public System.String get_Pattern();
```

Public Methods

```
public System.Web.Services.Discovery.DiscoveryReference GetDiscoveryReference(
   System.String filename
);
```

DiscoveryClientDocumentCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DiscoveryClientDocumentCollection
   extends System.Collections.DictionaryBase
```

Public Constructors

```
public DiscoveryClientDocumentCollection();
```

Public Properties

```
public System.Object get_Item(
   System.String url
);
public void set_Item(
   System.String url,
   System.Object value
);
public System.Collections.ICollection get_Keys();
public System.Collections.ICollection get_Values();
```

Public Methods

```
public void Add(
   System.String url,
   System.Object value
);
public boolean Contains(
   System.String url
);
public void Remove(
   System.String url
);
```

DiscoveryClientProtocol Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DiscoveryClientProtocol
   extends System.Web.Services.Protocols.HttpWebClientProtocol
```

Public Constructors

```
public DiscoveryClientProtocol();
```

Public Properties

```
public System.Collections.IList get_AdditionalInformation();
public System.Web.Services.Discovery.DiscoveryClientDocumentCollection get_Documents();
public System.Web.Services.Discovery.DiscoveryExceptionDictionary get_Errors();
public System.Web.Services.Discovery.DiscoveryClientReferenceCollection get_References();
```

Public Methods

```
public System.Web.Services.Discovery.DiscoveryDocument Discover(
   System.String url
);
public System.Web.Services.Discovery.DiscoveryDocument DiscoverAny(
   System.String url
);
public System.IO.Stream Download(
   System.String& url
);
public System.IO.Stream Download(
   System.String& url,
   System.String& contentType
);
public System.Web.Services.Discovery.DiscoveryClientResultCollection ReadAll(
   System.String topLevelFilename
);
public void ResolveAll();
public void ResolveOneLevel();
public System.Web.Services.Discovery.DiscoveryClientResultCollection WriteAll(
   System.String directory,
   System.String topLevelFilename
);
```

DiscoveryClientProtocol.DiscoveryClientResultsFile Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DiscoveryClientResultsFile
   extends System.Object
```

Public Constructors

```
public DiscoveryClientResultsFile();
```

Public Properties

```
public System.Web.Services.Discovery.DiscoveryClientResultCollection get_Results();
```

DiscoveryClientReferenceCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DiscoveryClientReferenceCollection
   extends System.Collections.DictionaryBase
```

Public Constructors

```
public DiscoveryClientReferenceCollection();
```

Public Properties

```
public System.Web.Services.Discovery.DiscoveryReference get_Item(
   System.String url
);
public void set_Item(
   System.String url,
   System.Web.Services.Discovery.DiscoveryReference value
);
public System.Collections.ICollection get_Keys();
public System.Collections.ICollection get_Values();
```

Public Methods

```
public void Add(
   System.Web.Services.Discovery.DiscoveryReference value
);
public void Add(
   System.String url,
   System.Web.Services.Discovery.DiscoveryReference value
);
public boolean Contains(
   System.String url
);
public void Remove(
   System.String url
);
```

DiscoveryClientResult Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DiscoveryClientResult
   extends System.Object
```

Public Constructors

```
public DiscoveryClientResult();
public DiscoveryClientResult(
   System.Type referenceType,
   System.String url,
   System.String filename
);
```

Public Properties

```
public System.String get_Filename();
public void set_Filename(
   System.String value
);
public System.String get_ReferenceTypeName();
public void set_ReferenceTypeName(
   System.String value
);
public System.String get_Url();
public void set_Url(
   System.String value
);
```

DiscoveryClientResultCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DiscoveryClientResultCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public DiscoveryClientResultCollection();
```

Public Properties

```
public System.Web.Services.Discovery.DiscoveryClientResult get_Item(
   int i
);
public void set_Item(
   int i,
   System.Web.Services.Discovery.DiscoveryClientResult value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Discovery.DiscoveryClientResult value
);
public boolean Contains(
   System.Web.Services.Discovery.DiscoveryClientResult value
);
public void Remove(
   System.Web.Services.Discovery.DiscoveryClientResult value
);
```

DiscoveryDocument Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Xml.Serialization.XmlRootAttribute(ElementName = "discovery", Namespace
= "http://schemas.xmlsoap.org/disco/", DataType = "", IsNullable = true) */
public class DiscoveryDocument
   extends System.Object
```

Public Constructors

```
public DiscoveryDocument();
```

Public Properties

```
public System.Collections.IList get_References();
```

Public Methods

```
public static boolean CanRead(
   System.Xml.XmlReader xmlReader
);
public static System.Web.Services.Discovery.DiscoveryDocument Read(
   System.IO.Stream stream
);
public static System.Web.Services.Discovery.DiscoveryDocument Read(
   System.IO.TextReader reader
);
public static System.Web.Services.Discovery.DiscoveryDocument Read(
   System.Xml.XmlReader xmlReader
);
public void Write(
   System.IO.Stream stream
);
public void Write(
   System.IO.TextWriter writer
);
public void Write(
   System.Xml.XmlWriter writer
);
```

Public Fields

```
public static System.String Namespace;
```

DiscoveryDocumentLinksPattern Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DiscoveryDocumentLinksPattern
   extends System.Web.Services.Discovery.DiscoverySearchPattern
```

Public Constructors

```
public DiscoveryDocumentLinksPattern();
```

Public Properties

```
public System.String get_Pattern();
```

Public Methods

```
public System.Web.Services.Discovery.DiscoveryReference GetDiscoveryReference(
   System.String filename
);
```

DiscoveryDocumentReference Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Xml.Serialization.XmlRootAttribute(ElementName = "discoveryRef", Namesp
ace = "http://schemas.xmlsoap.org/disco/", DataType = "", IsNullable = true) */
public class DiscoveryDocumentReference
   extends System.Web.Services.Discovery.DiscoveryReference
```

Public Constructors

```
public DiscoveryDocumentReference();
public DiscoveryDocumentReference(
   System.String href
);
```

Public Properties

```
public System.String get_DefaultFilename();
public System.Web.Services.Discovery.DiscoveryDocument get_Document();
public System.String get_Ref();
public void set_Ref(
   System.String value
);
public System.String get_Url();
public void set_Url(
   System.String value
);
```

Public Methods

```
public System.Object ReadDocument(
   System.IO.Stream stream
);
public void ResolveAll();
public void WriteDocument(
   System.Object document,
   System.IO.Stream stream
);
```

Protected Methods

```
protected void Resolve(
   System.String contentType,
   System.IO.Stream stream
);
```

DiscoveryDocumentSearchPattern Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DiscoveryDocumentSearchPattern
   extends System.Web.Services.Discovery.DiscoverySearchPattern
```

Public Constructors

```
public DiscoveryDocumentSearchPattern();
```

Public Properties

```
public System.String get_Pattern();
```

Public Methods

```
public System.Web.Services.Discovery.DiscoveryReference GetDiscoveryReference(
   System.String filename
);
```

DiscoveryExceptionDictionary Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DiscoveryExceptionDictionary
   extends System.Collections.DictionaryBase
```

Public Constructors

```
public DiscoveryExceptionDictionary();
```

Public Properties

```
public System.Exception get_Item(
   System.String url
);
public void set_Item(
   System.String url,
   System.Exception value
);
public System.Collections.ICollection get_Keys();
public System.Collections.ICollection get_Values();
```

Public Methods

```
public void Add(
   System.String url,
   System.Exception value
);
public boolean Contains(
   System.String url
);
public void Remove(
   System.String url
);
```

DiscoveryReference Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class DiscoveryReference
   extends System.Object
```

Public Properties

```
public System.Web.Services.Discovery.DiscoveryClientProtocol get_ClientProtocol();
public void set_ClientProtocol(
   System.Web.Services.Discovery.DiscoveryClientProtocol value
);
public System.String get_DefaultFilename();
public abstract System.String get_Url();
public abstract void set_Url(
   System.String value
);
```

Public Methods

```
public abstract System.Object ReadDocument(
   System.IO.Stream stream
);
public void Resolve();
public abstract void WriteDocument(
   System.Object document,
   System.IO.Stream stream
);
```

Protected Constructors

```
protected DiscoveryReference();
```

Protected Methods

```
protected static System.String FilenameFromUrl(
   System.String url
);
protected abstract void Resolve(
   System.String contentType,
   System.IO.Stream stream
);
```

DiscoveryReferenceCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DiscoveryReferenceCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public DiscoveryReferenceCollection();
```

Public Properties

```
public System.Web.Services.Discovery.DiscoveryReference get_Item(
   int i
);
public void set_Item(
   int i,
   System.Web.Services.Discovery.DiscoveryReference value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Discovery.DiscoveryReference value
);
public boolean Contains(
   System.Web.Services.Discovery.DiscoveryReference value
);
public void Remove(
   System.Web.Services.Discovery.DiscoveryReference value
);
```

DiscoveryRequestHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DiscoveryRequestHandler
   extends System.Object
   implements System.Web.IHttpHandler
```

Public Constructors

```
public DiscoveryRequestHandler();
```

Public Properties

```
public final boolean get_IsReusable();
```

Public Methods

```
public final void ProcessRequest(
   System.Web.HttpContext context
);
```

DiscoverySearchPattern Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class DiscoverySearchPattern
   extends System.Object
```

Public Properties

```
public abstract System.String get_Pattern();
```

Public Methods

```
public abstract System.Web.Services.Discovery.DiscoveryReference GetDiscoveryReference(
   System.String filename
);
```

Protected Constructors

```
protected DiscoverySearchPattern();
```

DynamicDiscoveryDocument Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Xml.Serialization.XmlRootAttribute(ElementName = "dynamicDiscovery", Na
mespace = "urn:schemas-dynamicdiscovery:disco.2000-03-17", DataType = "", IsNullable = true)
*/
public class DynamicDiscoveryDocument
   extends System.Object
```

Public Constructors

```
public DynamicDiscoveryDocument();
```

Public Properties

```
public System.Web.Services.Discovery.ExcludePathInfo[] get_ExcludePaths();
public void set_ExcludePaths(
   System.Web.Services.Discovery.ExcludePathInfo[] value
);
```

Public Methods

```
public static System.Web.Services.Discovery.DynamicDiscoveryDocument Load(
   System.IO.Stream stream
);
public void Write(
   System.IO.Stream stream
);
```

Public Fields

```
public static System.String Namespace;
```

ExcludePathInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ExcludePathInfo
   extends System.Object
```

Public Constructors

```
public ExcludePathInfo();
public ExcludePathInfo(
   System.String path
);
```

Public Properties

```
public System.String get_Path();
public void set_Path(
   System.String value
);
```

SchemaReference Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Xml.Serialization.XmlRootAttribute(ElementName = "schemaRef", Namespace
= "http://schemas.xmlsoap.org/disco/schema/", DataType = "", IsNullable = true) */
public class SchemaReference
   extends System.Web.Services.Discovery.DiscoveryReference
```

Public Constructors

```
public SchemaReference();
public SchemaReference(
   System.String url
);
```

Public Properties

```
public System.String get_DefaultFilename();
public System.String get_Ref();
public void set_Ref(
   System.String value
);
public System.Xml.Schema.XmlSchema get_Schema();
public System.String get_TargetNamespace();
public void set_TargetNamespace(
   System.String value
);
public System.String get_Url();
public void set_Url(
   System.String value
);
```

Public Methods

```
public System.Object ReadDocument(
   System.IO.Stream stream
);
public void WriteDocument(
   System.Object document,
   System.IO.Stream stream
);
```

Public Fields

```
public static System.String Namespace;
```

Protected Methods

```
protected void Resolve(
   System.String contentType,
   System.IO.Stream stream
);
```

SoapBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Xml.Serialization.XmlRootAttribute(ElementName = "soap", Namespace = "h
ttp://schemas.xmlsoap.org/disco/soap/", DataType = "", IsNullable = true) */
public class SoapBinding
   extends System.Object
```

Public Constructors

```
public SoapBinding();
```

Public Properties

```
public System.String get_Address();
public void set_Address(
   System.String value
);
public System.Xml.XmlQualifiedName get_Binding();
public void set_Binding(
   System.Xml.XmlQualifiedName value
);
```

Public Fields

```
public static System.String Namespace;
```

XmlSchemaSearchPattern Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaSearchPattern
   extends System.Web.Services.Discovery.DiscoverySearchPattern
```

Public Constructors

```
public XmlSchemaSearchPattern();
```

Public Properties

```
public System.String get_Pattern();
```

Public Methods

```
public System.Web.Services.Discovery.DiscoveryReference GetDiscoveryReference(
   System.String filename
);
```

System.Web.Services.Protocols Namespace (Visual J# Syntax)

Classes

AnyReturnReader

HtmlFormParameterReader

HtmlFormParameterWriter

HttpGetClientProtocol

HttpMethodAttribute

HttpPostClientProtocol

HttpSimpleClientProtocol

HttpWebClientProtocol

LogicalMethodInfo

MatchAttribute

MimeFormatter

MimeParameterReader

MimeParameterWriter

MimeReturnReader

NopReturnReader

PatternMatcher

SoapClientMessage

SoapDocumentMethodAttribute

SoapDocumentServiceAttribute

SoapException

SoapExtension

SoapExtensionAttribute

SoapHeader

SoapHeaderAttribute

SoapHeaderCollection

SoapHeaderException

SoapHttpClientProtocol

SoapMessage

SoapRpcMethodAttribute

SoapRpcServiceAttribute

SoapServerMessage

SoapUnknownHeader

TextReturnReader

UrlEncodedParameterWriter

UrlParameterReader

UrlParameterWriter

ValueCollectionParameterReader

WebClientAsyncResult

WebClientProtocol

WebServiceHandlerFactory

XmlReturnReader

Enumerations

LogicalMethodTypes

SoapHeaderDirection

SoapMessageStage

SoapParameterStyle

SoapServiceRoutingStyle

AnyReturnReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AnyReturnReader
   extends System.Web.Services.Protocols.MimeReturnReader
```

Public Constructors

```
public AnyReturnReader();
```

Public Methods

```
public System.Object GetInitializer(
   System.Web.Services.Protocols.LogicalMethodInfo methodInfo
);
public void Initialize(
   System.Object o
);
public System.Object Read(
   System.Net.WebResponse response,
   System.IO.Stream responseStream
);
```

HtmlFormParameterReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HtmlFormParameterReader
   extends System.Web.Services.Protocols.ValueCollectionParameterReader
```

Public Constructors

```
public HtmlFormParameterReader();
```

Public Methods

```
public System.Object[] Read(
   System.Web.HttpRequest request
);
```

HtmlFormParameterWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HtmlFormParameterWriter
   extends System.Web.Services.Protocols.UrlEncodedParameterWriter
```

Public Constructors

```
public HtmlFormParameterWriter();
```

Public Properties

```
public boolean get_UsesWriteRequest();
```

Public Methods

```
public void InitializeRequest(
   System.Net.WebRequest request,
   System.Object[] values
);
public void WriteRequest(
   System.IO.Stream requestStream,
   System.Object[] values
);
```

HttpGetClientProtocol Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpGetClientProtocol
   extends System.Web.Services.Protocols.HttpSimpleClientProtocol
```

Public Constructors

```
public HttpGetClientProtocol();
```

Protected Methods

```
protected System.Net.WebRequest GetWebRequest(
   System.Uri uri
);
```

HttpMethodAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = true) */
public class HttpMethodAttribute
   extends System.Attribute
```

Public Constructors

```
public HttpMethodAttribute();
public HttpMethodAttribute(
   System.Type returnFormatter,
   System.Type parameterFormatter
);
```

Public Properties

```
public System.Type get_ParameterFormatter();
public void set_ParameterFormatter(
   System.Type value
);
public System.Type get_ReturnFormatter();
public void set_ReturnFormatter(
   System.Type value
);
```

HttpPostClientProtocol Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HttpPostClientProtocol
   extends System.Web.Services.Protocols.HttpSimpleClientProtocol
```

Public Constructors

```
public HttpPostClientProtocol();
```

Protected Methods

```
protected System.Net.WebRequest GetWebRequest(
   System.Uri uri
);
```

HttpSimpleClientProtocol Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class HttpSimpleClientProtocol
   extends System.Web.Services.Protocols.HttpWebClientProtocol
```

Protected Constructors

```
protected HttpSimpleClientProtocol();
```

Protected Methods

```
protected System.IAsyncResult BeginInvoke(
   System.String methodName,
   System.String requestUrl,
   System.Object[] parameters,
   System.AsyncCallback callback,
   System.Object asyncState
);
protected System.Object EndInvoke(
   System.IAsyncResult asyncResult
);
protected System.Object Invoke(
   System.String methodName,
   System.String requestUrl,
   System.Object[] parameters
);
```

HttpWebClientProtocol Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class HttpWebClientProtocol
   extends System.Web.Services.Protocols.WebClientProtocol
```

Public Properties

```
public boolean get_AllowAutoRedirect();
public void set_AllowAutoRedirect(
   boolean value
);
public System.Security.Cryptography.X509Certificates.X509CertificateCollection get_ClientCert
ificates();
public System.Net.CookieContainer get_CookieContainer();
public void set_CookieContainer(
   System.Net.CookieContainer value
);
public System.Net.IWebProxy get_Proxy();
public void set_Proxy(
   System.Net.IWebProxy value
);
public System.String get_UserAgent();
public void set_UserAgent(
   System.String value
);
```

Protected Constructors

```
protected HttpWebClientProtocol();
```

Protected Methods

```
protected System.Net.WebRequest GetWebRequest(
   System.Uri uri
);
protected System.Net.WebResponse GetWebResponse(
   System.Net.WebRequest request
);
protected System.Net.WebResponse GetWebResponse(
   System.Net.WebRequest request,
   System.IAsyncResult result
);
```

LogicalMethodInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LogicalMethodInfo
   extends System.Object
```

Public Constructors

```
public LogicalMethodInfo(
   System.Reflection.MethodInfo methodInfo
);
```

Public Properties

```
public System.Reflection.ParameterInfo get_AsyncCallbackParameter();
public System.Reflection.ParameterInfo get_AsyncResultParameter();
public System.Reflection.ParameterInfo get_AsyncStateParameter();
public System.Reflection.MethodInfo get_BeginMethodInfo();
public System.Reflection.ICustomAttributeProvider get_CustomAttributeProvider();
public System.Type get_DeclaringType();
public System.Reflection.MethodInfo get_EndMethodInfo();
public System.Reflection.ParameterInfo[] get_InParameters();
public boolean get_IsAsync();
public boolean get_IsVoid();
public System.Reflection.MethodInfo get_MethodInfo();
public System.String get_Name();
public System.Reflection.ParameterInfo[] get_OutParameters();
public System.Reflection.ParameterInfo[] get_Parameters();
public System.Type get_ReturnType();
public System.Reflection.ICustomAttributeProvider get_ReturnTypeCustomAttributeProvider();
```

Public Methods

```
public System.IAsyncResult BeginInvoke(
   System.Object target,
   System.Object[] values,
   System.AsyncCallback callback,
   System.Object asyncState
);
public static System.Web.Services.Protocols.LogicalMethodInfo[] Create(
   System.Reflection.MethodInfo[] methodInfos
);
public static System.Web.Services.Protocols.LogicalMethodInfo[] Create(
   System.Reflection.MethodInfo[] methodInfos,
   System.Web.Services.Protocols.LogicalMethodTypes types
);
public System.Object[] EndInvoke(
   System.Object target,
   System.IAsyncResult asyncResult
);
public System.Object GetCustomAttribute(
   System.Type type
);
public System.Object[] GetCustomAttributes(
   System.Type type
);
```

```
public System.Object[] Invoke(
   System.Object target,
   System.Object[] values
);
public static boolean IsBeginMethod(
   System.Reflection.MethodInfo methodInfo
);
public static boolean IsEndMethod(
   System.Reflection.MethodInfo methodInfo
);
public System.String ToString();
```

LogicalMethodTypes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum LogicalMethodTypes
```

Members

Async

Sync

MatchAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class MatchAttribute
   extends System.Attribute
```

Public Constructors

```
public MatchAttribute(
   System.String pattern
);
```

Public Properties

```
public int get_Capture();
public void set_Capture(
   int value
);
public int get_Group();
public void set_Group(
   int value
);
public boolean get_IgnoreCase();
public void set_IgnoreCase(
   boolean value
);
public int get_MaxRepeats();
public void set_MaxRepeats(
   int value
);
public System.String get_Pattern();
public void set_Pattern(
   System.String value
);
```

MimeFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class MimeFormatter
   extends System.Object
```

Public Methods

```
public static System.Web.Services.Protocols.MimeFormatter CreateInstance(
   System.Type type,
   System.Object initializer
);
public abstract System.Object GetInitializer(
   System.Web.Services.Protocols.LogicalMethodInfo methodInfo
);
public static System.Object GetInitializer(
   System.Type type,
   System.Web.Services.Protocols.LogicalMethodInfo methodInfo
);
public System.Object[] GetInitializers(
   System.Web.Services.Protocols.LogicalMethodInfo[] methodInfos
);
public static System.Object[] GetInitializers(
   System.Type type,
   System.Web.Services.Protocols.LogicalMethodInfo[] methodInfos
);
public abstract void Initialize(
   System.Object initializer
);
```

Protected Constructors

```
protected MimeFormatter();
```

MimeParameterReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class MimeParameterReader
   extends System.Web.Services.Protocols.MimeFormatter
```

Public Methods

```
public abstract System.Object[] Read(
   System.Web.HttpRequest request
);
```

Protected Constructors

```
protected MimeParameterReader();
```

MimeParameterWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class MimeParameterWriter
   extends System.Web.Services.Protocols.MimeFormatter
```

Public Properties

```
public System.Text.Encoding get_RequestEncoding();
public void set_RequestEncoding(
   System.Text.Encoding value
);
public boolean get_UsesWriteRequest();
```

Public Methods

```
public System.String GetRequestUrl(
   System.String url,
   System.Object[] parameters
);
public void InitializeRequest(
   System.Net.WebRequest request,
   System.Object[] values
);
public void WriteRequest(
   System.IO.Stream requestStream,
   System.Object[] values
);
```

Protected Constructors

```
protected MimeParameterWriter();
```

MimeReturnReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class MimeReturnReader
   extends System.Web.Services.Protocols.MimeFormatter
```

Public Methods

```
public abstract System.Object Read(
   System.Net.WebResponse response,
   System.IO.Stream responseStream
);
```

Protected Constructors

```
protected MimeReturnReader();
```

NopReturnReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class NopReturnReader
   extends System.Web.Services.Protocols.MimeReturnReader
```

Public Constructors

```
public NopReturnReader();
```

Public Methods

```
public System.Object GetInitializer(
   System.Web.Services.Protocols.LogicalMethodInfo methodInfo
);
public void Initialize(
   System.Object initializer
);
public System.Object Read(
   System.Net.WebResponse response,
   System.IO.Stream responseStream
);
```

PatternMatcher Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PatternMatcher
   extends System.Object
```

Public Constructors

```
public PatternMatcher(
   System.Type type
);
```

Public Methods

```
public System.Object Match(
   System.String text
);
```

SoapClientMessage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapClientMessage
   extends System.Web.Services.Protocols.SoapMessage
```

Public Properties

```
public System.String get_Action();
public System.Web.Services.Protocols.SoapHttpClientProtocol get_Client();
public System.Web.Services.Protocols.LogicalMethodInfo get_MethodInfo();
public boolean get_OneWay();
public System.String get_Url();
```

Protected Methods

```
protected void EnsureInStage();
protected void EnsureOutStage();
```

SoapDocumentMethodAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = true) */
public class SoapDocumentMethodAttribute
   extends System.Attribute
```

Public Constructors

```
public SoapDocumentMethodAttribute();
public SoapDocumentMethodAttribute(
   System.String action
);
```

Public Properties

```
public System.String get_Action();
public void set_Action(
   System.String value
);
public System.String get_Binding();
public void set_Binding(
   System.String value
);
public boolean get_OneWay();
public void set_OneWay(
   boolean value
);
public System.Web.Services.Protocols.SoapParameterStyle get_ParameterStyle();
public void set_ParameterStyle(
   System.Web.Services.Protocols.SoapParameterStyle value
);
public System.String get_RequestElementName();
public void set_RequestElementName(
   System.String value
);
public System.String get_RequestNamespace();
public void set_RequestNamespace(
   System.String value
);
public System.String get_ResponseElementName();
public void set_ResponseElementName(
   System.String value
);
public System.String get_ResponseNamespace();
public void set_ResponseNamespace(
   System.String value
);
public System.Web.Services.Description.SoapBindingUse get_Use();
public void set_Use(
   System.Web.Services.Description.SoapBindingUse value
);
```

SoapDocumentServiceAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class SoapDocumentServiceAttribute
   extends System.Attribute
```

Public Constructors

```
public SoapDocumentServiceAttribute();
public SoapDocumentServiceAttribute(
   System.Web.Services.Description.SoapBindingUse use
);
public SoapDocumentServiceAttribute(
   System.Web.Services.Description.SoapBindingUse use,
   System.Web.Services.Protocols.SoapParameterStyle paramStyle
);
```

Public Properties

```
public System.Web.Services.Protocols.SoapParameterStyle get_ParameterStyle();
public void set_ParameterStyle(
   System.Web.Services.Protocols.SoapParameterStyle value
);
public System.Web.Services.Protocols.SoapServiceRoutingStyle get_RoutingStyle();
public void set_RoutingStyle(
   System.Web.Services.Protocols.SoapServiceRoutingStyle value
);
public System.Web.Services.Description.SoapBindingUse get_Use();
public void set_Use(
   System.Web.Services.Description.SoapBindingUse value
);
```

SoapException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapException
   extends System.SystemException
```

Public Constructors

```
public SoapException(
   System.String message,
   System.Xml.XmlQualifiedName code
);
public SoapException(
   System.String message,
   System.Xml.XmlQualifiedName code,
   System.Exception innerException
);
public SoapException(
   System.String message,
   System.Xml.XmlQualifiedName code,
   System.String actor
);
public SoapException(
   System.String message,
   System.Xml.XmlQualifiedName code,
   System.String actor,
   System.Exception innerException
);
public SoapException(
   System.String message,
   System.Xml.XmlQualifiedName code,
   System.String actor,
   System.Xml.XmlNode detail
);
public SoapException(
   System.String message,
   System.Xml.XmlQualifiedName code,
   System.String actor,
   System.Xml.XmlNode detail,
   System.Exception innerException
);
```

Public Properties

```
public System.String get_Actor();
public System.Xml.XmlQualifiedName get_Code();
public System.Xml.XmlNode get_Detail();
```

Public Fields

```
public static final System.Xml.XmlQualifiedName ClientFaultCode;
public static final System.Xml.XmlQualifiedName DetailElementName;
public static final System.Xml.XmlQualifiedName MustUnderstandFaultCode;
public static final System.Xml.XmlQualifiedName ServerFaultCode;
public static final System.Xml.XmlQualifiedName VersionMismatchFaultCode;
```

SoapExtension Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class SoapExtension
   extends System.Object
```

Public Methods

```
public System.IO.Stream ChainStream(
   System.IO.Stream stream
);
public abstract System.Object GetInitializer(
   System.Type serviceType
);
public abstract System.Object GetInitializer(
   System.Web.Services.Protocols.LogicalMethodInfo methodInfo,
   System.Web.Services.Protocols.SoapExtensionAttribute attribute
);
public abstract void Initialize(
   System.Object initializer
);
public abstract void ProcessMessage(
   System.Web.Services.Protocols.SoapMessage message
);
```

Protected Constructors

```
protected SoapExtension();
```

SoapExtensionAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class SoapExtensionAttribute
   extends System.Attribute
```

Public Properties

```
public abstract System.Type get_ExtensionType();
public abstract int get_Priority();
public abstract void set_Priority(
   int value
);
```

Protected Constructors

```
protected SoapExtensionAttribute();
```

SoapHeader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Xml.Serialization.SoapTypeAttribute(IncludeInSchema = false, TypeName =
"") */
/** @attribute System.Xml.Serialization.XmlTypeAttribute(IncludeInSchema = false, TypeName =
"") */
public abstract class SoapHeader
   extends System.Object
```

Public Properties

```
public System.String get_Actor();
public void set_Actor(
   System.String value
);
public boolean get_DidUnderstand();
public void set_DidUnderstand(
   boolean value
);
public System.String get_EncodedMustUnderstand();
public void set_EncodedMustUnderstand(
   System.String value
);
public boolean get_MustUnderstand();
public void set_MustUnderstand(
   boolean value
);
```

Protected Constructors

```
protected SoapHeader();
```

SoapHeaderAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = true, Inherit
ed = true) */
public class SoapHeaderAttribute
   extends System.Attribute
```

Public Constructors

```
public SoapHeaderAttribute(
   System.String memberName
);
```

Public Properties

```
public System.Web.Services.Protocols.SoapHeaderDirection get_Direction();
public void set_Direction(
   System.Web.Services.Protocols.SoapHeaderDirection value
);
public System.String get_MemberName();
public void set_MemberName(
   System.String value
);
public boolean get_Required();
public void set_Required(
   boolean value
);
```

SoapHeaderCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SoapHeaderCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public SoapHeaderCollection();
```

Public Properties

```
public System.Web.Services.Protocols.SoapHeader get_Item(
   int index
);
public void set_Item(
   int index,
   System.Web.Services.Protocols.SoapHeader value
);
```

Public Methods

```
public int Add(
   System.Web.Services.Protocols.SoapHeader header
);
public boolean Contains(
   System.Web.Services.Protocols.SoapHeader header
);
public void CopyTo(
   System.Web.Services.Protocols.SoapHeader[] array,
   int index
);
public int IndexOf(
   System.Web.Services.Protocols.SoapHeader header
);
public void Insert(
   int index,
   System.Web.Services.Protocols.SoapHeader header
);
public void Remove(
   System.Web.Services.Protocols.SoapHeader header
);
```

SoapHeaderDirection Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum SoapHeaderDirection
```

Members

In

InOut

Out

SoapHeaderException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapHeaderException
   extends System.Web.Services.Protocols.SoapException
```

Public Constructors

```
public SoapHeaderException(
   System.String message,
   System.Xml.XmlQualifiedName code
);
public SoapHeaderException(
   System.String message,
   System.Xml.XmlQualifiedName code,
   System.Exception innerException
);
public SoapHeaderException(
   System.String message,
   System.Xml.XmlQualifiedName code,
   System.String actor
);
public SoapHeaderException(
   System.String message,
   System.Xml.XmlQualifiedName code,
   System.String actor,
   System.Exception innerException
);
```

SoapHttpClientProtocol Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapHttpClientProtocol
   extends System.Web.Services.Protocols.HttpWebClientProtocol
```

Public Constructors

```
public SoapHttpClientProtocol();
```

Public Methods

```
public void Discover();
```

Protected Methods

```
protected System.IAsyncResult BeginInvoke(
   System.String methodName,
   System.Object[] parameters,
   System.AsyncCallback callback,
   System.Object asyncState
);
protected System.Object[] EndInvoke(
   System.IAsyncResult asyncResult
);
protected System.Net.WebRequest GetWebRequest(
   System.Uri uri
);
protected System.Object[] Invoke(
   System.String methodName,
   System.Object[] parameters
);
```

SoapMessage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class SoapMessage
   extends System.Object
```

Public Properties

```
public abstract System.String get_Action();
public System.String get_ContentType();
public void set_ContentType(
   System.String value
);
public System.Web.Services.Protocols.SoapException get_Exception();
public System.Web.Services.Protocols.SoapHeaderCollection get_Headers();
public abstract System.Web.Services.Protocols.LogicalMethodInfo get_MethodInfo();
public abstract boolean get_OneWay();
public System.Web.Services.Protocols.SoapMessageStage get_Stage();
public System.IO.Stream get_Stream();
public abstract System.String get_Url();
```

Public Methods

```
public System.Object GetInParameterValue(
   int index
);
public System.Object GetOutParameterValue(
   int index
);
public System.Object GetReturnValue();
```

Protected Methods

```
protected abstract void EnsureInStage();
protected abstract void EnsureOutStage();
protected void EnsureStage(
   System.Web.Services.Protocols.SoapMessageStage stage
);
```

SoapMessageStage Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SoapMessageStage
```

Members

AfterDeserialize

AfterSerialize

BeforeDeserialize

BeforeSerialize

SoapParameterStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SoapParameterStyle
```

Members

Bare

Default

Wrapped

SoapRpcMethodAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Method, AllowMultiple = false, Inheri
ted = true) */
public class SoapRpcMethodAttribute
   extends System.Attribute
```

Public Constructors

```
public SoapRpcMethodAttribute();
public SoapRpcMethodAttribute(
   System.String action
);
```

Public Properties

```
public System.String get_Action();
public void set_Action(
   System.String value
);
public System.String get_Binding();
public void set_Binding(
   System.String value
);
public boolean get_OneWay();
public void set_OneWay(
   boolean value
);
public System.String get_RequestElementName();
public void set_RequestElementName(
   System.String value
);
public System.String get_RequestNamespace();
public void set_RequestNamespace(
   System.String value
);
public System.String get_ResponseElementName();
public void set_ResponseElementName(
   System.String value
);
public System.String get_ResponseNamespace();
public void set_ResponseNamespace(
   System.String value
);
```

SoapRpcServiceAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class SoapRpcServiceAttribute
   extends System.Attribute
```

Public Constructors

```
public SoapRpcServiceAttribute();
```

Public Properties

```
public System.Web.Services.Protocols.SoapServiceRoutingStyle get_RoutingStyle();
public void set_RoutingStyle(
   System.Web.Services.Protocols.SoapServiceRoutingStyle value
);
```

SoapServerMessage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapServerMessage
   extends System.Web.Services.Protocols.SoapMessage
```

Public Properties

```
public System.String get_Action();
public System.Web.Services.Protocols.LogicalMethodInfo get_MethodInfo();
public boolean get_OneWay();
public System.Object get_Server();
public System.String get_Url();
```

Protected Methods

```
protected void EnsureInStage();
protected void EnsureOutStage();
```

SoapServiceRoutingStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SoapServiceRoutingStyle
```

Members

RequestElement

SoapAction

SoapUnknownHeader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapUnknownHeader
   extends System.Web.Services.Protocols.SoapHeader
```

Public Constructors

```
public SoapUnknownHeader();
```

Public Properties

```
public System.Xml.XmlElement get_Element();
public void set_Element(
   System.Xml.XmlElement value
);
```

TextReturnReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TextReturnReader
   extends System.Web.Services.Protocols.MimeReturnReader
```

Public Constructors

```
public TextReturnReader();
```

Public Methods

```
public System.Object GetInitializer(
   System.Web.Services.Protocols.LogicalMethodInfo methodInfo
);
public void Initialize(
   System.Object o
);
public System.Object Read(
   System.Net.WebResponse response,
   System.IO.Stream responseStream
);
```

UrlEncodedParameterWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class UrlEncodedParameterWriter
   extends System.Web.Services.Protocols.MimeParameterWriter
```

Public Properties

```
public System.Text.Encoding get_RequestEncoding();
public void set_RequestEncoding(
   System.Text.Encoding value
);
```

Public Methods

```
public System.Object GetInitializer(
   System.Web.Services.Protocols.LogicalMethodInfo methodInfo
);
public void Initialize(
   System.Object initializer
);
```

Protected Constructors

```
protected UrlEncodedParameterWriter();
```

Protected Methods

```
protected void Encode(
   System.IO.TextWriter writer,
   System.Object[] values
);
protected void Encode(
   System.IO.TextWriter writer,
   System.String name,
   System.Object value
);
```

UrlParameterReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UrlParameterReader
   extends System.Web.Services.Protocols.ValueCollectionParameterReader
```

Public Constructors

```
public UrlParameterReader();
```

Public Methods

```
public System.Object[] Read(
   System.Web.HttpRequest request
);
```

UrlParameterWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UrlParameterWriter
   extends System.Web.Services.Protocols.UrlEncodedParameterWriter
```

Public Constructors

```
public UrlParameterWriter();
```

Public Methods

```
public System.String GetRequestUrl(
   System.String url,
   System.Object[] parameters
);
```

ValueCollectionParameterReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class ValueCollectionParameterReader
   extends System.Web.Services.Protocols.MimeParameterReader
```

Public Methods

```
public System.Object GetInitializer(
   System.Web.Services.Protocols.LogicalMethodInfo methodInfo
);
public void Initialize(
   System.Object o
);
public static boolean IsSupported(
   System.Reflection.ParameterInfo paramInfo
);
public static boolean IsSupported(
   System.Web.Services.Protocols.LogicalMethodInfo methodInfo
);
```

Protected Constructors

```
protected ValueCollectionParameterReader();
```

Protected Methods

```
protected System.Object[] Read(
   System.Collections.Specialized.NameValueCollection collection
);
```

WebClientAsyncResult Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WebClientAsyncResult
   extends System.Object
   implements System.IAsyncResult
```

Public Properties

```
public final System.Object get_AsyncState();
public final System.Threading.WaitHandle get_AsyncWaitHandle();
public final boolean get_CompletedSynchronously();
public final boolean get_IsCompleted();
```

Public Methods

```
public void Abort();
```

WebClientProtocol Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class WebClientProtocol
   extends System.ComponentModel.Component
```

Public Properties

```
public System.String get_ConnectionGroupName();
public void set_ConnectionGroupName(
   System.String value
);
public System.Net.ICredentials get_Credentials();
public void set_Credentials(
   System.Net.ICredentials value
);
public boolean get_PreAuthenticate();
public void set_PreAuthenticate(
   boolean value
);
public System.Text.Encoding get_RequestEncoding();
public void set_RequestEncoding(
   System.Text.Encoding value
);
public int get_Timeout();
public void set_Timeout(
   int value
);
public System.String get_Url();
public void set_Url(
   System.String value
);
```

Public Methods

```
public void Abort();
```

Protected Constructors

```
protected WebClientProtocol();
```

Protected Methods

```
protected static void AddToCache(
   System.Type type,
   System.Object value
);
protected static System.Object GetFromCache(
   System.Type type
);
protected System.Net.WebRequest GetWebRequest(
   System.Uri uri
);
protected System.Net.WebResponse GetWebResponse(
```

```
   System.Net.WebRequest request
);
protected System.Net.WebResponse GetWebResponse(
   System.Net.WebRequest request,
   System.IAsyncResult result
);
```

WebServiceHandlerFactory Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WebServiceHandlerFactory
   extends System.Object
   implements System.Web.IHttpHandlerFactory
```

Public Constructors

```
public WebServiceHandlerFactory();
```

Public Methods

```
public final System.Web.IHttpHandler GetHandler(
   System.Web.HttpContext context,
   System.String verb,
   System.String url,
   System.String filePath
);
public final void ReleaseHandler(
   System.Web.IHttpHandler handler
);
```

XmlReturnReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlReturnReader
   extends System.Web.Services.Protocols.MimeReturnReader
```

Public Constructors

```
public XmlReturnReader();
```

Public Methods

```
public System.Object GetInitializer(
   System.Web.Services.Protocols.LogicalMethodInfo methodInfo
);
public System.Object[] GetInitializers(
   System.Web.Services.Protocols.LogicalMethodInfo[] methodInfos
);
public void Initialize(
   System.Object o
);
public System.Object Read(
   System.Net.WebResponse response,
   System.IO.Stream responseStream
);
```

System.Web.SessionState Namespace (Visual J# Syntax)

Classes

HttpSessionState

SessionStateModule

StateRuntime

Interfaces

IReadOnlySessionState

IRequiresSessionState

IStateRuntime

Enumerations

SessionStateMode

HttpSessionState Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HttpSessionState
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public int get_CodePage();
public void set_CodePage(
   int value
);
public System.Web.SessionState.HttpSessionState get_Contents();
public final int get_Count();
public boolean get_IsCookieless();
public boolean get_IsNewSession();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Object get_Item(
   int index
);
public System.Object get_Item(
   System.String name
);
public void set_Item(
   int index,
   System.Object value
);
public void set_Item(
   System.String name,
   System.Object value
);
public System.Collections.Specialized.NameObjectCollectionBase+KeysCollection get_Keys();
public int get_LCID();
public void set_LCID(
   int value
);
public System.Web.SessionState.SessionStateMode get_Mode();
public System.String get_SessionID();
public System.Web.HttpStaticObjectsCollection get_StaticObjects();
public final System.Object get_SyncRoot();
public int get_Timeout();
public void set_Timeout(
   int value
);
```

Public Methods

```
public void Abandon();
public void Add(
   System.String name,
   System.Object value
);
public void Clear();
public final void CopyTo(
   System.Array array,
```

```
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public void Remove(
   System.String name
);
public void RemoveAll();
public void RemoveAt(
   int index
);
```

IReadOnlySessionState Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IReadOnlySessionState
   implements System.Web.SessionState.IRequiresSessionState
```

IRequiresSessionState Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IRequiresSessionState
```

IStateRuntime Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "7297744b-e188-40bf-b7e9-
56698d25cf44") */
public interface IStateRuntime
```

Public Methods

```
void ProcessRequest(
   System.IntPtr tracker,
   int verb,
   System.String uri,
   int exclusive,
   int timeout,
   int lockCookieExists,
   int lockCookie,
   int contentLength,
   System.IntPtr content
);
void StopProcessing();
```

SessionStateMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SessionStateMode
```

Members

InProc

Off

SQLServer

StateServer

SessionStateModule Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SessionStateModule
   extends System.Object
   implements System.Web.IHttpModule
```

Public Constructors

```
public SessionStateModule();
```

Public Methods

```
public final void Dispose();
public final void Init(
   System.Web.HttpApplication app
);
```

Public Events

```
public void add_End(
   System.EventHandler value
);
public void remove_End(
   System.EventHandler value
);
public void add_Start(
   System.EventHandler value
);
public void remove_Start(
   System.EventHandler value
);
```

StateRuntime Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class StateRuntime
   extends System.Object
   implements System.Web.SessionState.IStateRuntime
```

Public Constructors

```
public StateRuntime();
```

Public Methods

```
public final void ProcessRequest(
   System.IntPtr tracker,
   int verb,
   System.String uri,
   int exclusive,
   int timeout,
   int lockCookieExists,
   int lockCookie,
   int contentLength,
   System.IntPtr content
);
public final void StopProcessing();
```

System.Web.UI Namespace (Visual J# Syntax)

Classes

AttributeCollection

BaseParser

BasePartialCachingControl

CompiledTemplateBuilder

ConstructorNeedsTagAttribute

Control

ControlBuilder

ControlBuilderAttribute

ControlCollection

CssStyleCollection

DataBinder

DataBinding

DataBindingCollection

DataBindingHandlerAttribute

DataBoundLiteralControl

DesignerDataBoundLiteralControl

DesignTimeParseData

DesignTimeTemplateParser

EmptyControlCollection

Html32TextWriter

HtmlTextWriter

ImageClickEventArgs

LiteralControl

LosFormatter

ObjectConverter

ObjectTagBuilder

Page

PageParser

Pair

ParseChildrenAttribute

PartialCachingAttribute

PartialCachingControl

PersistChildrenAttribute

PersistenceModeAttribute

PropertyConverter

RootBuilder

SimpleWebHandlerParser

StateBag

StateItem

StaticPartialCachingControl

TagPrefixAttribute

TemplateBuilder

TemplateContainerAttribute

TemplateControl

TemplateControlParser

TemplateParser

ToolboxDataAttribute

Triplet

UserControl

UserControlControlBuilder

ValidationPropertyAttribute

ValidatorCollection

WebServiceParser

Interfaces

IAttributeAccessor

IDataBindingsAccessor

INamingContainer

IParserAccessor

IPostBackDataHandler

IPostBackEventHandler

IStateManager

ITemplate

IUserControlDesignerAccessor

IValidator

Delegates

BuildMethod

BuildTemplateMethod

ImageClickEventHandler

RenderMethod

Enumerations

HtmlTextWriterAttribute

HtmlTextWriterStyle

HtmlTextWriterTag

OutputCacheLocation

PersistenceMode

AttributeCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class AttributeCollection
   extends System.Object
```

Public Constructors

```
public AttributeCollection(
   System.Web.UI.StateBag bag
);
```

Public Properties

```
public int get_Count();
public System.Web.UI.CssStyleCollection get_CssStyle();
public System.String get_Item(
   System.String key
);
public void set_Item(
   System.String key,
   System.String value
);
public System.Collections.ICollection get_Keys();
```

Public Methods

```
public void Add(
   System.String key,
   System.String value
);
public void AddAttributes(
   System.Web.UI.HtmlTextWriter writer
);
public void Clear();
public void Remove(
   System.String key
);
public void Render(
   System.Web.UI.HtmlTextWriter writer
);
```

BaseParser Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BaseParser
   extends System.Object
```

Public Constructors

```
public BaseParser();
```

BasePartialCachingControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public abstract class BasePartialCachingControl
   extends System.Web.UI.Control
```

Public Properties

```
public System.Web.Caching.CacheDependency get_Dependency();
public void set_Dependency(
   System.Web.Caching.CacheDependency value
);
```

Public Methods

```
public void Dispose();
```

Protected Constructors

```
protected BasePartialCachingControl();
```

Protected Methods

```
protected void OnInit(
   System.EventArgs e
);
protected void Render(
   System.Web.UI.HtmlTextWriter output
);
```

BuildMethod Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
System.Web.UI.Control BuildMethod();
```

BuildTemplateMethod Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void BuildTemplateMethod(
   System.Web.UI.Control control
);
```

CompiledTemplateBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CompiledTemplateBuilder
   extends System.Object
   implements System.Web.UI.ITemplate
```

Public Constructors

```
public CompiledTemplateBuilder(
   System.Web.UI.BuildTemplateMethod buildTemplateMethod
);
```

Public Methods

```
public final void InstantiateIn(
   System.Web.UI.Control container
);
```

ConstructorNeedsTagAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class ConstructorNeedsTagAttribute
   extends System.Attribute
```

Public Constructors

```
public ConstructorNeedsTagAttribute();
public ConstructorNeedsTagAttribute(
   boolean needsTag
);
```

Public Properties

```
public boolean get_NeedsTag();
```

Control Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemFilterAttribute(FilterString = "System.Web.UI
", FilterType = Require) */
/** @attribute System.ComponentModel.DesignerCategoryAttribute(Category = "Code") */
/** @attribute System.ComponentModel.Design.Serialization.DesignerSerializerAttribute(Seriali
zerTypeName = "Microsoft.VSDesigner.WebForms.ControlCodeDomSerializer, Microsoft.VSDesigner,
Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a", SerializerBaseTypeName
= "System.ComponentModel.Design.Serialization.CodeDomSerializer, System.Design, Version=1.0.3
300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.ControlDesigner, System.D
esign, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemType = System.Web.UI.Des
ign.WebControlToolboxItem, ToolboxItemTypeName = "System.Web.UI.Design.WebControlToolboxItem,
System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "ID") */
public class Control
   extends System.Object
   implements System.ComponentModel.IComponent, System.IDisposable, System.Web.UI.IParserAcce
ssor, System.Web.UI.IDataBindingsAccessor
```

Public Constructors

```
public Control();
```

Public Properties

```
public System.Web.UI.Control get_BindingContainer();
public System.String get_ClientID();
public System.Web.UI.ControlCollection get_Controls();
public boolean get_EnableViewState();
public void set_EnableViewState(
   boolean value
);
public System.String get_ID();
public void set_ID(
   System.String value
);
public System.Web.UI.Control get_NamingContainer();
public System.Web.UI.Page get_Page();
public void set_Page(
   System.Web.UI.Page value
);
public System.Web.UI.Control get_Parent();
public final System.ComponentModel.ISite get_Site();
public final void set_Site(
   System.ComponentModel.ISite value
);
public System.String get_TemplateSourceDirectory();
public System.String get_UniqueID();
public boolean get_Visible();
public void set_Visible(
   boolean value
);
```

Public Methods

```
public void DataBind();
public void Dispose();
public System.Web.UI.Control FindControl(
   System.String id
);
public boolean HasControls();
public void RenderControl(
   System.Web.UI.HtmlTextWriter writer
);
public System.String ResolveUrl(
   System.String relativeUrl
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public void SetRenderMethodDelegate(
   System.Web.UI.RenderMethod renderMethod
);
```

Public Events

```
public void add_DataBinding(
   System.EventHandler value
);
public void remove_DataBinding(
   System.EventHandler value
);
public final void add_Disposed(
   System.EventHandler value
);
public final void remove_Disposed(
   System.EventHandler value
);
public void add_Init(
   System.EventHandler value
);
public void remove_Init(
   System.EventHandler value
);
public void add_Load(
   System.EventHandler value
);
public void remove_Load(
   System.EventHandler value
);
public void add_PreRender(
   System.EventHandler value
);
public void remove_PreRender(
   System.EventHandler value
);
public void add_Unload(
   System.EventHandler value
);
public void remove_Unload(
   System.EventHandler value
);
```

Protected Properties

```
protected boolean get_ChildControlsCreated();
protected void set_ChildControlsCreated(
   boolean value
```

```
);
protected System.Web.HttpContext get_Context();
protected System.ComponentModel.EventHandlerList get_Events();
protected boolean get_HasChildViewState();
protected boolean get_IsTrackingViewState();
protected System.Web.UI.StateBag get_ViewState();
protected boolean get_ViewStateIgnoresCase();
```

Protected Methods

```
protected void AddedControl(
   System.Web.UI.Control control,
   int index
);
protected void AddParsedSubObject(
   System.Object obj
);
protected void BuildProfileTree(
   System.String parentId,
   boolean calcViewState
);
protected void ClearChildViewState();
protected void CreateChildControls();
protected System.Web.UI.ControlCollection CreateControlCollection();
protected void EnsureChildControls();
protected System.Web.UI.Control FindControl(
   System.String id,
   int pathOffset
);
protected boolean IsLiteralContent();
protected void LoadViewState(
   System.Object savedState
);
protected System.String MapPathSecure(
   System.String virtualPath
);
protected boolean OnBubbleEvent(
   System.Object source,
   System.EventArgs args
);
protected void OnDataBinding(
   System.EventArgs e
);
protected void OnInit(
   System.EventArgs e
);
protected void OnLoad(
   System.EventArgs e
);
protected void OnPreRender(
   System.EventArgs e
);
protected void OnUnload(
   System.EventArgs e
);
protected void RaiseBubbleEvent(
   System.Object source,
   System.EventArgs args
);
protected void RemovedControl(
   System.Web.UI.Control control
);
protected void Render(
   System.Web.UI.HtmlTextWriter writer
);
protected void RenderChildren(
```

```
    System.Web.UI.HtmlTextWriter writer
);
protected System.Object SaveViewState();
protected void TrackViewState();
```

ControlBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ControlBuilder
   extends System.Object
```

Public Constructors

```
public ControlBuilder();
```

Public Properties

```
public System.Type get_ControlType();
public boolean get_HasAspCode();
public System.String get_ID();
public void set_ID(
   System.String value
);
public System.Type get_NamingContainerType();
public System.String get_TagName();
```

Public Methods

```
public boolean AllowWhitespaceLiterals();
public void AppendLiteralString(
   System.String s
);
public void AppendSubBuilder(
   System.Web.UI.ControlBuilder subBuilder
);
public void CloseControl();
public static System.Web.UI.ControlBuilder CreateBuilderFromType(
   System.Web.UI.TemplateParser parser,
   System.Web.UI.ControlBuilder parentBuilder,
   System.Type type,
   System.String tagName,
   System.String id,
   System.Collections.IDictionary attribs,
   int line,
   System.String sourceFileName
);
public System.Type GetChildControlType(
   System.String tagName,
   System.Collections.IDictionary attribs
);
public boolean HasBody();
public boolean HtmlDecodeLiterals();
public void Init(
   System.Web.UI.TemplateParser parser,
   System.Web.UI.ControlBuilder parentBuilder,
   System.Type type,
   System.String tagName,
   System.String id,
   System.Collections.IDictionary attribs
);
public boolean NeedsTagInnerText();
```

```
public void OnAppendToParentBuilder(
   System.Web.UI.ControlBuilder parentBuilder
);
public void SetTagInnerText(
   System.String text
);
```

Protected Properties

```
protected boolean get_FChildrenAsProperties();
protected boolean get_FIsNonParserAccessor();
protected boolean get_InDesigner();
protected System.Web.UI.TemplateParser get_Parser();
```

ControlBuilderAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class ControlBuilderAttribute
   extends System.Attribute
```

Public Constructors

```
public ControlBuilderAttribute(
   System.Type builderType
);
```

Public Properties

```
public System.Type get_BuilderType();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.Web.UI.ControlBuilderAttribute Default;
```

ControlCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ControlCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public ControlCollection(
   System.Web.UI.Control owner
);
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Web.UI.Control get_Item(
   int index
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public void Add(
   System.Web.UI.Control child
);
public void AddAt(
   int index,
   System.Web.UI.Control child
);
public void Clear();
public boolean Contains(
   System.Web.UI.Control c
);
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Web.UI.Control value
);
public void Remove(
   System.Web.UI.Control value
);
public void RemoveAt(
   int index
);
```

Protected Properties

```
protected System.Web.UI.Control get_Owner();
```

CssStyleCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CssStyleCollection
   extends System.Object
```

Public Properties

```
public int get_Count();
public System.String get_Item(
   System.String key
);
public void set_Item(
   System.String key,
   System.String value
);
public System.Collections.ICollection get_Keys();
```

Public Methods

```
public void Add(
   System.String key,
   System.String value
);
public void Clear();
public void Remove(
   System.String key
);
```

DataBinder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataBinder
   extends System.Object
```

Public Constructors

```
public DataBinder();
```

Public Methods

```
public static System.Object Eval(
   System.Object container,
   System.String expression
);
public static System.String Eval(
   System.Object container,
   System.String expression,
   System.String format
);
public static System.Object GetIndexedPropertyValue(
   System.Object container,
   System.String expr
);
public static System.String GetIndexedPropertyValue(
   System.Object container,
   System.String propName,
   System.String format
);
public static System.Object GetPropertyValue(
   System.Object container,
   System.String propName
);
public static System.String GetPropertyValue(
   System.Object container,
   System.String propName,
   System.String format
);
```

DataBinding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataBinding
   extends System.Object
```

Public Constructors

```
public DataBinding(
   System.String propertyName,
   System.Type propertyType,
   System.String expression
);
```

Public Properties

```
public System.String get_Expression();
public void set_Expression(
   System.String value
);
public System.String get_PropertyName();
public System.Type get_PropertyType();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
```

DataBindingCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DataBindingCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public DataBindingCollection();
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Web.UI.DataBinding get_Item(
   System.String propertyName
);
public System.String[] get_RemovedBindings();
public final System.Object get_SyncRoot();
```

Public Methods

```
public void Add(
   System.Web.UI.DataBinding binding
);
public void Clear();
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public void Remove(
   System.String propertyName
);
public void Remove(
   System.Web.UI.DataBinding binding
);
public void Remove(
   System.String propertyName,
   boolean addToRemovedList
);
```

DataBindingHandlerAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class DataBindingHandlerAttribute
   extends System.Attribute
```

Public Constructors

```
public DataBindingHandlerAttribute();
public DataBindingHandlerAttribute(
   System.String typeName
);
public DataBindingHandlerAttribute(
   System.Type type
);
```

Public Properties

```
public System.String get_HandlerTypeName();
```

Public Fields

```
public static final System.Web.UI.DataBindingHandlerAttribute Default;
```

DataBoundLiteralControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class DataBoundLiteralControl
   extends System.Web.UI.Control
```

Public Constructors

```
public DataBoundLiteralControl(
   int staticLiteralsCount,
   int dataBoundLiteralCount
);
```

Public Properties

```
public System.String get_Text();
```

Public Methods

```
public void SetDataBoundString(
   int index,
   System.String s
);
public void SetStaticString(
   int index,
   System.String s
);
```

Protected Methods

```
protected System.Web.UI.ControlCollection CreateControlCollection();
protected void LoadViewState(
   System.Object savedState
);
protected void Render(
   System.Web.UI.HtmlTextWriter output
);
protected System.Object SaveViewState();
```

DesignerDataBoundLiteralControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.DataBindingHandlerAttribute(HandlerTypeName = "System.Web.UI.Des
ign.TextDataBindingHandler, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToke
n=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class DesignerDataBoundLiteralControl
   extends System.Web.UI.Control
```

Public Constructors

```
public DesignerDataBoundLiteralControl();
```

Public Properties

```
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Protected Methods

```
protected System.Web.UI.ControlCollection CreateControlCollection();
protected void LoadViewState(
   System.Object savedState
);
protected void Render(
   System.Web.UI.HtmlTextWriter output
);
protected System.Object SaveViewState();
```

DesignTimeParseData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DesignTimeParseData
   extends System.Object
```

Public Constructors

```
public DesignTimeParseData(
   System.ComponentModel.Design.IDesignerHost designerHost,
   System.String parseText
);
```

Public Properties

```
public System.EventHandler get_DataBindingHandler();
public void set_DataBindingHandler(
   System.EventHandler value
);
public System.ComponentModel.Design.IDesignerHost get_DesignerHost();
public System.String get_DocumentUrl();
public void set_DocumentUrl(
   System.String value
);
public System.String get_ParseText();
```

DesignTimeTemplateParser Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DesignTimeTemplateParser
   extends System.Object
```

Public Methods

```
public static System.Web.UI.Control ParseControl(
   System.Web.UI.DesignTimeParseData data
);
public static System.Web.UI.ITemplate ParseTemplate(
   System.Web.UI.DesignTimeParseData data
);
```

EmptyControlCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EmptyControlCollection
   extends System.Web.UI.ControlCollection
```

Public Constructors

```
public EmptyControlCollection(
   System.Web.UI.Control owner
);
```

Public Methods

```
public void Add(
   System.Web.UI.Control child
);
public void AddAt(
   int index,
   System.Web.UI.Control child
);
```

Html32TextWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Html32TextWriter
   extends System.Web.UI.HtmlTextWriter
```

Public Constructors

```
public Html32TextWriter(
   System.IO.TextWriter writer
);
public Html32TextWriter(
   System.IO.TextWriter writer,
   System.String tabString
);
```

Public Methods

```
public void RenderBeginTag(
   System.Web.UI.HtmlTextWriterTag tagKey
);
public void RenderEndTag();
```

Protected Properties

```
protected System.Collections.Stack get_FontStack();
```

Protected Methods

```
protected System.String GetTagName(
   System.Web.UI.HtmlTextWriterTag tagKey
);
protected boolean OnStyleAttributeRender(
   System.String name,
   System.String value,
   System.Web.UI.HtmlTextWriterStyle key
);
protected boolean OnTagRender(
   System.String name,
   System.Web.UI.HtmlTextWriterTag key
);
protected System.String RenderAfterContent();
protected System.String RenderAfterTag();
protected System.String RenderBeforeContent();
protected System.String RenderBeforeTag();
```

HtmlTextWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HtmlTextWriter
   extends System.IO.TextWriter
```

Public Constructors

```
public HtmlTextWriter(
   System.IO.TextWriter writer
);
public HtmlTextWriter(
   System.IO.TextWriter writer,
   System.String tabString
);
```

Public Properties

```
public System.Text.Encoding get_Encoding();
public int get_Indent();
public void set_Indent(
   int value
);
public System.IO.TextWriter get_InnerWriter();
public void set_InnerWriter(
   System.IO.TextWriter value
);
public System.String get_NewLine();
public void set_NewLine(
   System.String value
);
```

Public Methods

```
public void AddAttribute(
   System.String name,
   System.String value
);
public void AddAttribute(
   System.Web.UI.HtmlTextWriterAttribute key,
   System.String value
);
public void AddAttribute(
   System.String name,
   System.String value,
   boolean fEndode
);
public void AddAttribute(
   System.Web.UI.HtmlTextWriterAttribute key,
   System.String value,
   boolean fEncode
);
public void AddStyleAttribute(
   System.String name,
   System.String value
);
```

```
public void AddStyleAttribute(
   System.Web.UI.HtmlTextWriterStyle key,
   System.String value
);
public void Close();
public void Flush();
public void RenderBeginTag(
   System.String tagName
);
public void RenderBeginTag(
   System.Web.UI.HtmlTextWriterTag tagKey
);
public void RenderEndTag();
public void Write(
   boolean value
);
public void Write(
   char value
);
public void Write(
   char[] buffer
);
public void Write(
   double value
);
public void Write(
   int value
);
public void Write(
   long value
);
public void Write(
   float value
);
public void Write(
   System.Object value
);
public void Write(
   System.String s
);
public void Write(
   System.String format,
   System.Object arg0
);
public void Write(
   System.String format,
   System.Object[] arg
);
public void Write(
   char[] buffer,
   int index,
   int count
);
public void Write(
   System.String format,
   System.Object arg0,
   System.Object arg1
);
public void WriteAttribute(
   System.String name,
   System.String value
);
public void WriteAttribute(
   System.String name,
   System.String value,
   boolean fEncode
);
public void WriteBeginTag(
```

```
   System.String tagName
);
public void WriteEndTag(
   System.String tagName
);
public void WriteFullBeginTag(
   System.String tagName
);
public void WriteLine();
public void WriteLine(
   boolean value
);
public void WriteLine(
   char value
);
public void WriteLine(
   char[] buffer
);
public void WriteLine(
   double value
);
public void WriteLine(
   int value
);
public void WriteLine(
   long value
);
public void WriteLine(
   float value
);
public void WriteLine(
   System.Object value
);
public void WriteLine(
   System.String s
);
public void WriteLine(
   System.String format,
   System.Object arg0
);
public void WriteLine(
   System.String format,
   System.Object[] arg
);
public void WriteLine(
   char[] buffer,
   int index,
   int count
);
public void WriteLine(
   System.String format,
   System.Object arg0,
   System.Object arg1
);
public void WriteLineNoTabs(
   System.String s
);
public void WriteStyleAttribute(
   System.String name,
   System.String value
);
public void WriteStyleAttribute(
   System.String name,
   System.String value,
   boolean fEncode
);
```

Public Fields

```
public static System.String DefaultTabString;
public static char DoubleQuoteChar;
public static System.String EndTagLeftChars;
public static char EqualsChar;
public static System.String EqualsDoubleQuoteString;
public static System.String SelfClosingChars;
public static System.String SelfClosingTagEnd;
public static char SemicolonChar;
public static char SingleQuoteChar;
public static char SlashChar;
public static char SpaceChar;
public static char StyleEqualsChar;
public static char TagLeftChar;
public static char TagRightChar;
```

Protected Properties

```
protected System.Web.UI.HtmlTextWriterTag get_TagKey();
protected void set_TagKey(
   System.Web.UI.HtmlTextWriterTag value
);
protected System.String get_TagName();
protected void set_TagName(
   System.String value
);
```

Protected Methods

```
protected void AddAttribute(
   System.String name,
   System.String value,
   System.Web.UI.HtmlTextWriterAttribute key
);
protected void AddStyleAttribute(
   System.String name,
   System.String value,
   System.Web.UI.HtmlTextWriterStyle key
);
protected System.String EncodeAttributeValue(
   System.String value,
   boolean fEncode
);
protected System.String EncodeAttributeValue(
   System.Web.UI.HtmlTextWriterAttribute attrKey,
   System.String value
);
protected System.String EncodeUrl(
   System.String url
);
protected void FilterAttributes();
protected System.Web.UI.HtmlTextWriterAttribute GetAttributeKey(
   System.String attrName
);
protected System.String GetAttributeName(
   System.Web.UI.HtmlTextWriterAttribute attrKey
);
protected System.Web.UI.HtmlTextWriterStyle GetStyleKey(
   System.String styleName
);
protected System.String GetStyleName(
   System.Web.UI.HtmlTextWriterStyle styleKey
```

```
);
protected System.Web.UI.HtmlTextWriterTag GetTagKey(
   System.String tagName
);
protected System.String GetTagName(
   System.Web.UI.HtmlTextWriterTag tagKey
);
protected boolean IsAttributeDefined(
   System.Web.UI.HtmlTextWriterAttribute key
);
protected boolean IsAttributeDefined(
   System.Web.UI.HtmlTextWriterAttribute key,
   System.String& value
);
protected boolean IsStyleAttributeDefined(
   System.Web.UI.HtmlTextWriterStyle key
);
protected boolean IsStyleAttributeDefined(
   System.Web.UI.HtmlTextWriterStyle key,
   System.String& value
);
protected boolean OnAttributeRender(
   System.String name,
   System.String value,
   System.Web.UI.HtmlTextWriterAttribute key
);
protected boolean OnStyleAttributeRender(
   System.String name,
   System.String value,
   System.Web.UI.HtmlTextWriterStyle key
);
protected boolean OnTagRender(
   System.String name,
   System.Web.UI.HtmlTextWriterTag key
);
protected void OutputTabs();
protected System.String PopEndTag();
protected void PushEndTag(
   System.String endTag
);
protected static void RegisterAttribute(
   System.String name,
   System.Web.UI.HtmlTextWriterAttribute key
);
protected static void RegisterStyle(
   System.String name,
   System.Web.UI.HtmlTextWriterStyle key
);
protected static void RegisterTag(
   System.String name,
   System.Web.UI.HtmlTextWriterTag key
);
protected System.String RenderAfterContent();
protected System.String RenderAfterTag();
protected System.String RenderBeforeContent();
protected System.String RenderBeforeTag();
```

HtmlTextWriterAttribute Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum HtmlTextWriterAttribute
```

Members

Accesskey

Align

Alt

Background

Bgcolor

Border

Bordercolor

Cellpadding

Cellspacing

Checked

Class

Cols

Colspan

Disabled

For

Height

Href

Id

Maxlength

Multiple

Name

Nowrap

Onchange

Onclick

ReadOnly

Rows

Rowspan

Rules

Selected

Size

Src

Style

Tabindex

Target

Title

Type

Valign

Value

Width

Wrap

HtmlTextWriterStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum HtmlTextWriterStyle
```

Members

BackgroundColor

BackgroundImage

BorderCollapse

BorderColor

BorderStyle

BorderWidth

Color

FontFamily

FontSize

FontStyle

FontWeight

Height

TextDecoration

Width

HtmlTextWriterTag Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum HtmlTextWriterTag
```

Members

A

Acronym

Address

Area

B

Base

Basefont

Bdo

Bgsound

Big

Blockquote

Body

Br

Button

Caption

Center

Cite

Code

Col

Colgroup

Dd

Del

Dfn

Dir

Div

Dl

Dt

Em

Embed

Fieldset

Font

Form

Frame

Frameset

H1

H2

H3

H4

H5

H6

Head

Hr

Html

I

Iframe

Img

Input

Ins

Isindex

Kbd

Label

Legend

Li

Link

Map

Marquee

Menu

Meta

Nobr

Noframes

Noscript

Object

Ol

Option

P

Param

Pre

Q

Rt

Ruby

S

Samp

Script

Select

Small

Span

Strike

Strong

Style

Sub

Sup

Table

Tbody

Td

Textarea

Tfoot

Th

Thead

Title

Tr

Tt

U

Ul

Unknown

Var

Wbr

Xml

IAttributeAccessor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IAttributeAccessor
```

Public Methods

```
System.String GetAttribute(
   System.String key
);
void SetAttribute(
   System.String key,
   System.String value
);
```

IDataBindingsAccessor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDataBindingsAccessor
```

Public Properties

```
System.Web.UI.DataBindingCollection get_DataBindings();
boolean get_HasDataBindings();
```

ImageClickEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ImageClickEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ImageClickEventArgs(
   int x,
   int y
);
```

Public Fields

```
public int X;
public int Y;
```

ImageClickEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ImageClickEventHandler(
   System.Object sender,
   System.Web.UI.ImageClickEventArgs e
);
```

INamingContainer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface INamingContainer
```

IParserAccessor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IParserAccessor
```

Public Methods

```
void AddParsedSubObject(
   System.Object obj
);
```

IPostBackDataHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IPostBackDataHandler
```

Public Methods

```
boolean LoadPostData(
   System.String postDataKey,
   System.Collections.Specialized.NameValueCollection postCollection
);
void RaisePostDataChangedEvent();
```

IPostBackEventHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IPostBackEventHandler
```

Public Methods

```
void RaisePostBackEvent(
   System.String eventArgument
);
```

IStateManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IStateManager
```

Public Properties

```
boolean get_IsTrackingViewState();
```

Public Methods

```
void LoadViewState(
   System.Object state
);
System.Object SaveViewState();
void TrackViewState();
```

ITemplate Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ITemplate
```

Public Methods

```
void InstantiateIn(
   System.Web.UI.Control container
);
```

IUserControlDesignerAccessor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IUserControlDesignerAccessor
```

Public Properties

```
System.String get_InnerText();
void set_InnerText(
   System.String value
);
System.String get_TagName();
void set_TagName(
   System.String value
);
```

IValidator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IValidator
```

Public Properties

```
System.String get_ErrorMessage();
void set_ErrorMessage(
   System.String value
);
boolean get_IsValid();
void set_IsValid(
   boolean value
);
```

Public Methods

```
void Validate();
```

LiteralControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class LiteralControl
   extends System.Web.UI.Control
```

Public Constructors

```
public LiteralControl();
public LiteralControl(
   System.String text
);
```

Public Properties

```
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Protected Methods

```
protected System.Web.UI.ControlCollection CreateControlCollection();
protected void Render(
   System.Web.UI.HtmlTextWriter output
);
```

LosFormatter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LosFormatter
   extends System.Object
```

Public Constructors

```
public LosFormatter();
```

Public Methods

```
public System.Object Deserialize(
   System.IO.Stream stream
);
public System.Object Deserialize(
   System.IO.TextReader input
);
public System.Object Deserialize(
   System.String input
);
public void Serialize(
   System.IO.Stream stream,
   System.Object value
);
public void Serialize(
   System.IO.TextWriter output,
   System.Object value
);
```

ObjectConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ObjectConverter
   extends System.Object
```

Public Constructors

```
public ObjectConverter();
```

Public Methods

```
public static System.Object ConvertValue(
   System.Object value,
   System.Type toType,
   System.String formatString
);
```

ObjectTagBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ObjectTagBuilder
   extends System.Web.UI.ControlBuilder
```

Public Constructors

```
public ObjectTagBuilder();
```

Public Methods

```
public void AppendLiteralString(
   System.String s
);
public void AppendSubBuilder(
   System.Web.UI.ControlBuilder subBuilder
);
public void Init(
   System.Web.UI.TemplateParser parser,
   System.Web.UI.ControlBuilder parentBuilder,
   System.Type type,
   System.String tagName,
   System.String id,
   System.Collections.IDictionary attribs
);
```

OutputCacheLocation Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum OutputCacheLocation
```

Members

Any

Client

Downstream

None

Server

Page Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IRootDesigner, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77
a5c561934e089", DesignerTypeName = "Microsoft.VSDesigner.WebForms.WebFormDesigner, Microsoft.
VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DesignerCategoryAttribute(Category = "ASPXCodeBehind") *
/
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Load") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.ComponentModel.Design.Serialization.RootDesignerSerializerAttribute(Rel
oadable = true, SerializerTypeName = "Microsoft.VSDesigner.WebForms.RootCodeDomSerializer, Mi
crosoft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a", Se
rializerBaseTypeName = "System.ComponentModel.Design.Serialization.CodeDomSerializer, System.
Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class Page
   extends System.Web.UI.TemplateControl
   implements System.Web.IHttpHandler
```

Public Constructors

```
public Page();
```

Public Properties

```
public System.Web.HttpApplicationState get_Application();
public System.Web.Caching.Cache get_Cache();
public System.String get_ClientTarget();
public void set_ClientTarget(
   System.String value
);
public boolean get_EnableViewState();
public void set_EnableViewState(
   boolean value
);
public System.String get_ErrorPage();
public void set_ErrorPage(
   System.String value
);
public System.String get_ID();
public void set_ID(
   System.String value
);
public boolean get_IsPostBack();
public final boolean get_IsReusable();
public boolean get_IsValid();
public System.Web.HttpRequest get_Request();
public System.Web.HttpResponse get_Response();
public System.Web.HttpServerUtility get_Server();
public System.Web.SessionState.HttpSessionState get_Session();
public boolean get_SmartNavigation();
public void set_SmartNavigation(
   boolean value
);
public System.Web.TraceContext get_Trace();
public System.Security.Principal.IPrincipal get_User();
public System.Web.UI.ValidatorCollection get_Validators();
```

```
public boolean get_Visible();
public void set_Visible(
   boolean value
);
```

Public Methods

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public void DesignerInitialize();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public System.String GetPostBackClientEvent(
   System.Web.UI.Control control,
   System.String argument
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public System.String GetPostBackClientHyperlink(
   System.Web.UI.Control control,
   System.String argument
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public System.String GetPostBackEventReference(
   System.Web.UI.Control control
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public System.String GetPostBackEventReference(
   System.Web.UI.Control control,
   System.String argument
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public int GetTypeHashCode();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public boolean IsClientScriptBlockRegistered(
   System.String key
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public boolean IsStartupScriptRegistered(
   System.String key
);
public System.String MapPath(
   System.String virtualPath
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public final void ProcessRequest(
   System.Web.HttpContext context
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public void RegisterArrayDeclaration(
   System.String arrayName,
   System.String arrayValue
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public void RegisterClientScriptBlock(
   System.String key,
   System.String script
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public void RegisterHiddenField(
   System.String hiddenFieldName,
   System.String hiddenFieldInitialValue
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public void RegisterOnSubmitStatement(
   System.String key,
   System.String script
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
```

```
public void RegisterRequiresPostBack(
   System.Web.UI.Control control
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public void RegisterRequiresRaiseEvent(
   System.Web.UI.IPostBackEventHandler control
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public void RegisterStartupScript(
   System.String key,
   System.String script
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public void RegisterViewStateHandler();
public void Validate();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public void VerifyRenderingInServerForm(
   System.Web.UI.Control control
);
```

Protected Properties

```
protected void set_AspCompatMode(
   boolean value
);
protected void set_Buffer(
   boolean value
);
protected void set_CodePage(
   int value
);
protected void set_ContentType(
   System.String value
);
protected System.Web.HttpContext get_Context();
protected void set_Culture(
   System.String value
);
protected boolean get_EnableViewStateMac();
protected void set_EnableViewStateMac(
   boolean value
);
protected void set_FileDependencies(
   System.Collections.ArrayList value
);
protected void set_LCID(
   int value
);
protected void set_ResponseEncoding(
   System.String value
);
protected void set_TraceEnabled(
   boolean value
);
protected void set_TraceModeValue(
   System.Web.TraceMode value
);
protected void set_TransactionMode(
   int value
);
protected void set_UICulture(
   System.String value
);
```

Protected Methods

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
protected System.IAsyncResult AspCompatBeginProcessRequest(
   System.Web.HttpContext context,
   System.AsyncCallback cb,
   System.Object extraData
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
protected void AspCompatEndProcessRequest(
   System.IAsyncResult result
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected System.Web.UI.HtmlTextWriter CreateHtmlTextWriter(
   System.IO.TextWriter tw
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected System.Collections.Specialized.NameValueCollection DeterminePostBackMode();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
protected void InitOutputCache(
   int duration,
   System.String varyByHeader,
   System.String varyByCustom,
   System.Web.UI.OutputCacheLocation location,
   System.String varyByParam
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected System.Object LoadPageStateFromPersistenceMedium();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RaisePostBackEvent(
   System.Web.UI.IPostBackEventHandler sourceControl,
   System.String eventArgument
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void SavePageStateToPersistenceMedium(
   System.Object viewState
);
```

Protected Fields

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
protected static System.String postEventArgumentID;
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
protected static System.String postEventSourceID;
```

PageParser Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PageParser
   extends System.Web.UI.TemplateControlParser
```

Public Constructors

```
public PageParser();
```

Public Methods

```
public static System.Web.IHttpHandler GetCompiledPageInstance(
   System.String virtualPath,
   System.String inputFile,
   System.Web.HttpContext context
);
```

Protected Methods

```
protected System.Type CompileIntoType();
```

Pair Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Pair
   extends System.Object
```

Public Constructors

```
public Pair();
public Pair(
   System.Object x,
   System.Object y
);
```

Public Fields

```
public System.Object First;
public System.Object Second;
```

ParseChildrenAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class ParseChildrenAttribute
   extends System.Attribute
```

Public Constructors

```
public ParseChildrenAttribute();
public ParseChildrenAttribute(
   boolean childrenAsProperties
);
public ParseChildrenAttribute(
   boolean childrenAsProperties,
   System.String defaultProperty
);
```

Public Properties

```
public boolean get_ChildrenAsProperties();
public void set_ChildrenAsProperties(
   boolean value
);
public System.String get_DefaultProperty();
public void set_DefaultProperty(
   System.String value
);
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.Web.UI.ParseChildrenAttribute Default;
```

PartialCachingAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class PartialCachingAttribute
   extends System.Attribute
```

Public Constructors

```
public PartialCachingAttribute(
   int duration
);
public PartialCachingAttribute(
   int duration,
   System.String varyByParams,
   System.String varyByControls,
   System.String varyByCustom
);
```

Public Properties

```
public int get_Duration();
public System.String get_VaryByControls();
public System.String get_VaryByCustom();
public System.String get_VaryByParams();
```

PartialCachingControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PartialCachingControl
   extends System.Web.UI.BasePartialCachingControl
```

Public Properties

```
public System.Web.UI.Control get_CachedControl();
```

PersistChildrenAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class PersistChildrenAttribute
   extends System.Attribute
```

Public Constructors

```
public PersistChildrenAttribute(
   boolean persist
);
```

Public Properties

```
public boolean get_Persist();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.Web.UI.PersistChildrenAttribute Default;
public static final System.Web.UI.PersistChildrenAttribute No;
public static final System.Web.UI.PersistChildrenAttribute Yes;
```

PersistenceMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PersistenceMode
```

Members

Attribute

EncodedInnerDefaultProperty

InnerDefaultProperty

InnerProperty

PersistenceModeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = All, AllowMultiple = false, Inherited
= true) */
public class PersistenceModeAttribute
   extends System.Attribute
```

Public Constructors

```
public PersistenceModeAttribute(
   System.Web.UI.PersistenceMode mode
);
```

Public Properties

```
public System.Web.UI.PersistenceMode get_Mode();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.Web.UI.PersistenceModeAttribute Attribute;
public static final System.Web.UI.PersistenceModeAttribute Default;
public static final System.Web.UI.PersistenceModeAttribute EncodedInnerDefaultProperty;
public static final System.Web.UI.PersistenceModeAttribute InnerDefaultProperty;
public static final System.Web.UI.PersistenceModeAttribute InnerProperty;
```

PropertyConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PropertyConverter
   extends System.Object
```

Public Methods

```
public static System.Object EnumFromString(
   System.Type enumType,
   System.String value
);
public static System.String EnumToString(
   System.Type enumType,
   System.Object enumValue
);
public static System.Object ObjectFromString(
   System.Type objType,
   System.Reflection.MemberInfo propertyInfo,
   System.String value
);
```

RenderMethod Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void RenderMethod(
   System.Web.UI.HtmlTextWriter output,
   System.Web.UI.Control container
);
```

RootBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RootBuilder
   extends System.Web.UI.TemplateBuilder
```

Public Constructors

```
public RootBuilder(
   System.Web.UI.TemplateParser parser
);
```

Public Methods

```
public System.Type GetChildControlType(
   System.String tagName,
   System.Collections.IDictionary attribs
);
```

SimpleWebHandlerParser Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class SimpleWebHandlerParser
   extends System.Object
```

Protected Constructors

```
protected SimpleWebHandlerParser(
   System.Web.HttpContext context,
   System.String virtualPath,
   System.String physicalPath
);
```

Protected Properties

```
protected abstract System.String get_DefaultDirectiveName();
```

Protected Methods

```
protected System.Type GetCompiledTypeFromCache();
```

StateBag Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class StateBag
   extends System.Object
   implements System.Web.UI.IStateManager, System.Collections.IDictionary, System.Collections
.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public StateBag();
public StateBag(
   boolean ignoreCase
);
```

Public Properties

```
public final int get_Count();
public System.Object get_Item(
   System.String key
);
public void set_Item(
   System.String key,
   System.Object value
);
public final System.Collections.ICollection get_Keys();
public final System.Collections.ICollection get_Values();
```

Public Methods

```
public System.Web.UI.StateItem Add(
   System.String key,
   System.Object value
);
public final void Clear();
public final System.Collections.IDictionaryEnumerator GetEnumerator();
public boolean IsItemDirty(
   System.String key
);
public void Remove(
   System.String key
);
public void SetItemDirty(
   System.String key,
   boolean dirty
);
```

StateItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class StateItem
   extends System.Object
```

Public Properties

```
public boolean get_IsDirty();
public void set_IsDirty(
   boolean value
);
public System.Object get_Value();
public void set_Value(
   System.Object value
);
```

StaticPartialCachingControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class StaticPartialCachingControl
   extends System.Web.UI.BasePartialCachingControl
```

Public Constructors

```
public StaticPartialCachingControl(
   System.String ctrlID,
   System.String guid,
   int duration,
   System.String varyByParams,
   System.String varyByControls,
   System.String varyByCustom,
   System.Web.UI.BuildMethod buildMethod
);
```

Public Methods

```
public static void BuildCachedControl(
   System.Web.UI.Control parent,
   System.String ctrlID,
   System.String guid,
   int duration,
   System.String varyByParams,
   System.String varyByControls,
   System.String varyByCustom,
   System.Web.UI.BuildMethod buildMethod
);
```

TagPrefixAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = true, Inher
ited = true) */
public class TagPrefixAttribute
   extends System.Attribute
```

Public Constructors

```
public TagPrefixAttribute(
   System.String namespaceName,
   System.String tagPrefix
);
```

Public Properties

```
public System.String get_NamespaceName();
public System.String get_TagPrefix();
```

TemplateBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TemplateBuilder
   extends System.Web.UI.ControlBuilder
   implements System.Web.UI.ITemplate
```

Public Constructors

```
public TemplateBuilder();
```

Public Properties

```
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Methods

```
public void Init(
   System.Web.UI.TemplateParser parser,
   System.Web.UI.ControlBuilder parentBuilder,
   System.Type type,
   System.String tagName,
   System.String ID,
   System.Collections.IDictionary attribs
);
public void InstantiateIn(
   System.Web.UI.Control container
);
public boolean NeedsTagInnerText();
public void SetTagInnerText(
   System.String text
);
```

TemplateContainerAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, AllowMultiple = false, Inhe
rited = true) */
public class TemplateContainerAttribute
   extends System.Attribute
```

Public Constructors

```
public TemplateContainerAttribute(
   System.Type containerType
);
```

Public Properties

```
public System.Type get_ContainerType();
```

TemplateControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class TemplateControl
   extends System.Web.UI.Control
   implements System.Web.UI.INamingContainer
```

Public Methods

```
public System.Web.UI.Control LoadControl(
   System.String virtualPath
);
public System.Web.UI.ITemplate LoadTemplate(
   System.String virtualPath
);
public System.Web.UI.Control ParseControl(
   System.String content
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public static System.Object ReadStringResource(
   System.Type t
);
```

Public Events

```
public void add_AbortTransaction(
   System.EventHandler value
);
public void remove_AbortTransaction(
   System.EventHandler value
);
public void add_CommitTransaction(
   System.EventHandler value
);
public void remove_CommitTransaction(
   System.EventHandler value
);
public void add_Error(
   System.EventHandler value
);
public void remove_Error(
   System.EventHandler value
);
```

Protected Constructors

```
protected TemplateControl();
```

Protected Properties

```
protected int get_AutoHandlers();
protected void set_AutoHandlers(
   int value
);
```

```
protected boolean get_SupportAutoEvents();
```

Protected Methods

```
protected void Construct();
protected System.Web.UI.LiteralControl CreateResourceBasedLiteralControl(
   int offset,
   int size,
   boolean fAsciiOnly
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
protected void FrameworkInitialize();
protected void OnAbortTransaction(
   System.EventArgs e
);
protected void OnCommitTransaction(
   System.EventArgs e
);
protected void OnError(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
protected void SetStringResourcePointer(
   System.Object stringResourcePointer,
   int maxResourceOffset
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
protected void WriteUTF8ResourceString(
   System.Web.UI.HtmlTextWriter output,
   int offset,
   int size,
   boolean fAsciiOnly
);
```

TemplateControlParser Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class TemplateControlParser
   extends System.Web.UI.TemplateParser
```

Protected Constructors

```
protected TemplateControlParser();
```

TemplateParser Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class TemplateParser
   extends System.Web.UI.BaseParser
```

Protected Methods

```
protected abstract System.Type CompileIntoType();
```

ToolboxDataAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class ToolboxDataAttribute
   extends System.Attribute
```

Public Constructors

```
public ToolboxDataAttribute(
   System.String data
);
```

Public Properties

```
public System.String get_Data();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public boolean IsDefaultAttribute();
```

Public Fields

```
public static final System.Web.UI.ToolboxDataAttribute Default;
```

Triplet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Triplet
   extends System.Object
```

Public Constructors

```
public Triplet();
public Triplet(
   System.Object x,
   System.Object y
);
public Triplet(
   System.Object x,
   System.Object y,
   System.Object z
);
```

Public Fields

```
public System.Object First;
public System.Object Second;
public System.Object Third;
```

UserControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IRootDesigner, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77
a5c561934e089", DesignerTypeName = "Microsoft.VSDesigner.WebForms.WebFormDesigner, Microsoft.
VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.Web.UI.ParseChildrenAttribute(ChildrenAsProperties = true, DefaultPrope
rty = "") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c5
61934e089", DesignerTypeName = "System.Web.UI.Design.UserControlDesigner, System.Design, Vers
ion=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.ComponentModel.Design.Serialization.RootDesignerSerializerAttribute(Rel
oadable = true, SerializerTypeName = "Microsoft.VSDesigner.WebForms.RootCodeDomSerializer, Mi
crosoft.VSDesigner, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a", Se
rializerBaseTypeName = "System.ComponentModel.Design.Serialization.CodeDomSerializer, System.
Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Load") */
/** @attribute System.ComponentModel.DesignerCategoryAttribute(Category = "ASPXCodeBehind") *
/
/** @attribute System.Web.UI.ControlBuilderAttribute(BuilderType = System.Web.UI.UserControlC
ontrolBuilder) */
public class UserControl
   extends System.Web.UI.TemplateControl
   implements System.Web.UI.IAttributeAccessor, System.Web.UI.IUserControlDesignerAccessor
```

Public Constructors

```
public UserControl();
```

Public Properties

```
public System.Web.HttpApplicationState get_Application();
public System.Web.UI.AttributeCollection get_Attributes();
public System.Web.Caching.Cache get_Cache();
public boolean get_IsPostBack();
public System.Web.HttpRequest get_Request();
public System.Web.HttpResponse get_Response();
public System.Web.HttpServerUtility get_Server();
public System.Web.SessionState.HttpSessionState get_Session();
public System.Web.TraceContext get_Trace();
```

Public Methods

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public void DesignerInitialize();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public void InitializeAsUserControl(
   System.Web.UI.Page page
);
public System.String MapPath(
   System.String virtualPath
);
```

Protected Methods

```
protected void LoadViewState(
   System.Object savedState
);
protected void OnInit(
   System.EventArgs e
);
protected System.Object SaveViewState();
```

UserControlControlBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UserControlControlBuilder
   extends System.Web.UI.ControlBuilder
```

Public Constructors

```
public UserControlControlBuilder();
```

Public Methods

```
public boolean NeedsTagInnerText();
public void SetTagInnerText(
   System.String text
);
```

ValidationPropertyAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = true) */
public class ValidationPropertyAttribute
   extends System.Attribute
```

Public Constructors

```
public ValidationPropertyAttribute(
   System.String name
);
```

Public Properties

```
public System.String get_Name();
```

ValidatorCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ValidatorCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public ValidatorCollection();
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Web.UI.IValidator get_Item(
   int index
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public void Add(
   System.Web.UI.IValidator validator
);
public boolean Contains(
   System.Web.UI.IValidator validator
);
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public void Remove(
   System.Web.UI.IValidator validator
);
```

WebServiceParser Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WebServiceParser
   extends System.Web.UI.SimpleWebHandlerParser
```

Public Methods

```
public static System.Type GetCompiledType(
   System.String inputFile,
   System.Web.HttpContext context
);
```

Protected Properties

```
protected System.String get_DefaultDirectiveName();
```

System.Web.UI.Design Namespace (Visual J# Syntax)

Classes

CalendarDataBindingHandler

ColorBuilder

ControlDesigner

ControlParser

ControlPersister

DataBindingCollectionConverter

DataBindingCollectionEditor

DataBindingHandler

DataBindingValueUIHandler

DataFieldConverter

DataMemberConverter

DataSourceConverter

DesignTimeData

HtmlControlDesigner

HtmlIntrinsicControlDesigner

HyperLinkDataBindingHandler

ImageUrlEditor

ReadWriteControlDesigner

TemplatedControlDesigner

TemplateEditingService

TemplateEditingVerb

TextControlDesigner

TextDataBindingHandler

UrlBuilder

UrlEditor

UserControlDesigner

WebControlToolboxItem

XmlFileEditor

XmlUrlEditor

XslUrlEditor

Interfaces

IControlDesignerBehavior

IDataSourceProvider

IHtmlControlDesignerBehavior

ITemplateEditingFrame

ITemplateEditingService

IWebFormReferenceManager

IWebFormsBuilderUIService

IWebFormsDocumentService

Enumerations

UrlBuilderOptions

CalendarDataBindingHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CalendarDataBindingHandler
   extends System.Web.UI.Design.DataBindingHandler
```

Public Constructors

```
public CalendarDataBindingHandler();
```

Public Methods

```
public void DataBindControl(
   System.ComponentModel.Design.IDesignerHost designerHost,
   System.Web.UI.Control control
);
```

ColorBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ColorBuilder
   extends System.Object
```

Public Methods

```
public static System.String BuildColor(
   System.ComponentModel.IComponent component,
   System.Windows.Forms.Control owner,
   System.String initialColor
);
```

ControlDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ControlDesigner
   extends System.Web.UI.Design.HtmlControlDesigner
```

Public Constructors

```
public ControlDesigner();
```

Public Properties

```
public boolean get_AllowResize();
public boolean get_DesignTimeHtmlRequiresLoadComplete();
public System.String get_ID();
public void set_ID(
   System.String value
);
public boolean get_IsDirty();
public void set_IsDirty(
   boolean value
);
public boolean get_ReadOnly();
public void set_ReadOnly(
   boolean value
);
```

Public Methods

```
public System.String GetDesignTimeHtml();
public System.String GetPersistInnerHtml();
public void Initialize(
   System.ComponentModel.IComponent component
);
public boolean IsPropertyBound(
   System.String propName
);
public void OnComponentChanged(
   System.Object sender,
   System.ComponentModel.Design.ComponentChangedEventArgs ce
);
public void RaiseResizeEvent();
public void UpdateDesignTimeHtml();
```

Protected Properties

```
protected System.Object get_DesignTimeElementView();
```

Protected Methods

```
protected System.String CreatePlaceHolderDesignTimeHtml();
protected System.String CreatePlaceHolderDesignTimeHtml(
```

```
   System.String instruction
);
protected System.String GetEmptyDesignTimeHtml();
protected System.String GetErrorDesignTimeHtml(
   System.Exception e
);
protected void OnBehaviorAttached();
protected void OnBindingsCollectionChanged(
   System.String propName
);
protected void OnControlResize();
protected void PreFilterProperties(
   System.Collections.IDictionary properties
);
```

ControlParser Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ControlParser
   extends System.Object
```

Public Methods

```
public static System.Web.UI.Control ParseControl(
   System.ComponentModel.Design.IDesignerHost designerHost,
   System.String controlText
);
public static System.Web.UI.Control ParseControl(
   System.ComponentModel.Design.IDesignerHost designerHost,
   System.String controlText,
   System.String directives
);
public static System.Web.UI.ITemplate ParseTemplate(
   System.ComponentModel.Design.IDesignerHost designerHost,
   System.String templateText
);
public static System.Web.UI.ITemplate ParseTemplate(
   System.ComponentModel.Design.IDesignerHost designerHost,
   System.String templateText,
   System.String directives
);
```

ControlPersister Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ControlPersister
   extends System.Object
```

Public Methods

```
public static System.String PersistControl(
   System.Web.UI.Control control
);
public static System.String PersistControl(
   System.Web.UI.Control control,
   System.ComponentModel.Design.IDesignerHost host
);
public static void PersistControl(
   System.IO.TextWriter sw,
   System.Web.UI.Control control
);
public static void PersistControl(
   System.IO.TextWriter sw,
   System.Web.UI.Control control,
   System.ComponentModel.Design.IDesignerHost host
);
public static System.String PersistInnerProperties(
   System.Object component,
   System.ComponentModel.Design.IDesignerHost host
);
public static void PersistInnerProperties(
   System.IO.TextWriter sw,
   System.Object component,
   System.ComponentModel.Design.IDesignerHost host
);
```

DataBindingCollectionConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataBindingCollectionConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public DataBindingCollectionConverter();
```

Public Methods

```
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

DataBindingCollectionEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataBindingCollectionEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public DataBindingCollectionEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

DataBindingHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class DataBindingHandler
   extends System.Object
```

Public Methods

```
public abstract void DataBindControl(
   System.ComponentModel.Design.IDesignerHost designerHost,
   System.Web.UI.Control control
);
```

Protected Constructors

```
protected DataBindingHandler();
```

DataBindingValueUIHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataBindingValueUIHandler
   extends System.Object
```

Public Constructors

```
public DataBindingValueUIHandler();
```

Public Methods

```
public void OnGetUIValueItem(
   System.ComponentModel.ITypeDescriptorContext context,
   System.ComponentModel.PropertyDescriptor propDesc,
   System.Collections.ArrayList valueUIItemList
);
```

DataFieldConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataFieldConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public DataFieldConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

DataMemberConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataMemberConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public DataMemberConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

DataSourceConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataSourceConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public DataSourceConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

DesignTimeData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DesignTimeData
   extends System.Object
```

Public Methods

```
public static System.Data.DataTable CreateDummyDataTable();
public static System.Data.DataTable CreateSampleDataTable(
   System.Collections.IEnumerable referenceData
);
public static System.ComponentModel.PropertyDescriptorCollection GetDataFields(
   System.Collections.IEnumerable dataSource
);
public static System.Collections.IEnumerable GetDataMember(
   System.ComponentModel.IListSource dataSource,
   System.String dataMember
);
public static System.String[] GetDataMembers(
   System.Object dataSource
);
public static System.Collections.IEnumerable GetDesignTimeDataSource(
   System.Data.DataTable dataTable,
   int minimumRows
);
public static System.Object GetSelectedDataSource(
   System.ComponentModel.IComponent component,
   System.String dataSource
);
public static System.Collections.IEnumerable GetSelectedDataSource(
   System.ComponentModel.IComponent component,
   System.String dataSource,
   System.String dataMember
);
```

Public Fields

```
public static final System.EventHandler DataBindingHandler;
```

HtmlControlDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HtmlControlDesigner
   extends System.ComponentModel.Design.ComponentDesigner
```

Public Constructors

```
public HtmlControlDesigner();
```

Public Properties

```
public System.Web.UI.Design.IHtmlControlDesignerBehavior get_Behavior();
public void set_Behavior(
   System.Web.UI.Design.IHtmlControlDesignerBehavior value
);
public System.Web.UI.DataBindingCollection get_DataBindings();
public boolean get_ShouldCodeSerialize();
public void set_ShouldCodeSerialize(
   boolean value
);
```

Public Methods

```
public void OnSetParent();
```

Protected Properties

```
protected System.Object get_DesignTimeElement();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void OnBehaviorAttached();
protected void OnBehaviorDetaching();
protected void OnBindingsCollectionChanged(
   System.String propName
);
protected void PreFilterEvents(
   System.Collections.IDictionary events
);
protected void PreFilterProperties(
   System.Collections.IDictionary properties
);
```

HtmlIntrinsicControlDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HtmlIntrinsicControlDesigner
   extends System.Web.UI.Design.HtmlControlDesigner
```

Public Constructors

```
public HtmlIntrinsicControlDesigner();
```

HyperLinkDataBindingHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HyperLinkDataBindingHandler
   extends System.Web.UI.Design.DataBindingHandler
```

Public Constructors

```
public HyperLinkDataBindingHandler();
```

Public Methods

```
public void DataBindControl(
   System.ComponentModel.Design.IDesignerHost designerHost,
   System.Web.UI.Control control
);
```

IControlDesignerBehavior Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IControlDesignerBehavior
```

Public Properties

```
System.Object get_DesignTimeElementView();
System.String get_DesignTimeHtml();
void set_DesignTimeHtml(
   System.String value
);
```

Public Methods

```
void OnTemplateModeChanged();
```

IDataSourceProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDataSourceProvider
```

Public Methods

```
System.Collections.IEnumerable GetResolvedSelectedDataSource();
System.Object GetSelectedDataSource();
```

IHtmlControlDesignerBehavior Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IHtmlControlDesignerBehavior
```

Public Properties

```
System.Web.UI.Design.HtmlControlDesigner get_Designer();
void set_Designer(
   System.Web.UI.Design.HtmlControlDesigner value
);
System.Object get_DesignTimeElement();
```

Public Methods

```
System.Object GetAttribute(
   System.String attribute,
   boolean ignoreCase
);
System.Object GetStyleAttribute(
   System.String attribute,
   boolean designTimeOnly,
   boolean ignoreCase
);
void RemoveAttribute(
   System.String attribute,
   boolean ignoreCase
);
void RemoveStyleAttribute(
   System.String attribute,
   boolean designTimeOnly,
   boolean ignoreCase
);
void SetAttribute(
   System.String attribute,
   System.Object value,
   boolean ignoreCase
);
void SetStyleAttribute(
   System.String attribute,
   boolean designTimeOnly,
   System.Object value,
   boolean ignoreCase
);
```

ImageUrlEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ImageUrlEditor
   extends System.Web.UI.Design.UrlEditor
```

Public Constructors

```
public ImageUrlEditor();
```

Protected Properties

```
protected System.String get_Caption();
protected System.String get_Filter();
```

ITemplateEditingFrame Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ITemplateEditingFrame
   implements System.IDisposable
```

Public Properties

```
System.Web.UI.WebControls.Style get_ControlStyle();
int get_InitialHeight();
void set_InitialHeight(
   int value
);
int get_InitialWidth();
void set_InitialWidth(
   int value
);
System.String get_Name();
System.String[] get_TemplateNames();
System.Web.UI.WebControls.Style[] get_TemplateStyles();
System.Web.UI.Design.TemplateEditingVerb get_Verb();
void set_Verb(
   System.Web.UI.Design.TemplateEditingVerb value
);
```

Public Methods

```
void Close(
   boolean saveChanges
);
void Open();
void Resize(
   int width,
   int height
);
void Save();
void UpdateControlName(
   System.String newName
);
```

ITemplateEditingService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ITemplateEditingService
```

Public Properties

```
boolean get_SupportsNestedTemplateEditing();
```

Public Methods

```
System.Web.UI.Design.ITemplateEditingFrame CreateFrame(
   System.Web.UI.Design.TemplatedControlDesigner designer,
   System.String frameName,
   System.String[] templateNames
);
System.Web.UI.Design.ITemplateEditingFrame CreateFrame(
   System.Web.UI.Design.TemplatedControlDesigner designer,
   System.String frameName,
   System.String[] templateNames,
   System.Web.UI.WebControls.Style controlStyle,
   System.Web.UI.WebControls.Style[] templateStyles
);
System.String GetContainingTemplateName(
   System.Web.UI.Control control
);
```

IWebFormReferenceManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IWebFormReferenceManager
```

Public Methods

```
System.Type GetObjectType(
   System.String tagPrefix,
   System.String typeName
);
System.String GetRegisterDirectives();
System.String GetTagPrefix(
   System.Type objectType
);
```

IWebFormsBuilderUIService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IWebFormsBuilderUIService
```

Public Methods

```
System.String BuildColor(
   System.Windows.Forms.Control owner,
   System.String initialColor
);
System.String BuildUrl(
   System.Windows.Forms.Control owner,
   System.String initialUrl,
   System.String baseUrl,
   System.String caption,
   System.String filter,
   System.Web.UI.Design.UrlBuilderOptions options
);
```

IWebFormsDocumentService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IWebFormsDocumentService
```

Public Properties

```
System.String get_DocumentUrl();
boolean get_IsLoading();
```

Public Methods

```
System.Object CreateDiscardableUndoUnit();
void DiscardUndoUnit(
   System.Object discardableUndoUnit
);
void EnableUndo(
   boolean enable
);
void UpdateSelection();
```

Public Events

```
void add_LoadComplete(
   System.EventHandler value
);
void remove_LoadComplete(
   System.EventHandler value
);
```

ReadWriteControlDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ReadWriteControlDesigner
   extends System.Web.UI.Design.ControlDesigner
```

Public Constructors

```
public ReadWriteControlDesigner();
```

Public Methods

```
public void OnComponentChanged(
   System.Object sender,
   System.ComponentModel.Design.ComponentChangedEventArgs ce
);
```

Protected Methods

```
protected void MapPropertyToStyle(
   System.String propName,
   System.Object varPropValue
);
protected void OnBehaviorAttached();
```

TemplatedControlDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class TemplatedControlDesigner
   extends System.Web.UI.Design.ControlDesigner
```

Public Constructors

```
public TemplatedControlDesigner();
```

Public Properties

```
public System.Web.UI.Design.ITemplateEditingFrame get_ActiveTemplateEditingFrame();
public boolean get_CanEnterTemplateMode();
public boolean get_InTemplateMode();
```

Public Methods

```
public void EnterTemplateMode(
   System.Web.UI.Design.ITemplateEditingFrame newTemplateEditingFrame
);
public void ExitTemplateMode(
   boolean fSwitchingTemplates,
   boolean fNested,
   boolean fSave
);
public System.String GetPersistInnerHtml();
public System.String GetTemplateContainerDataItemProperty(
   System.String templateName
);
public System.Collections.IEnumerable GetTemplateContainerDataSource(
   System.String templateName
);
public abstract System.String GetTemplateContent(
   System.Web.UI.Design.ITemplateEditingFrame editingFrame,
   System.String templateName,
   boolean allowEditing
);
public System.Web.UI.Design.TemplateEditingVerb[] GetTemplateEditingVerbs();
public System.Type GetTemplatePropertyParentType(
   System.String templateName
);
public void OnComponentChanged(
   System.Object sender,
   System.ComponentModel.Design.ComponentChangedEventArgs ce
);
public void OnSetParent();
public abstract void SetTemplateContent(
   System.Web.UI.Design.ITemplateEditingFrame editingFrame,
   System.String templateName,
   System.String templateContent
);
public void UpdateDesignTimeHtml();
```

Protected Properties

```
protected boolean get_HidePropertiesInTemplateMode();
```

Protected Methods

```
protected abstract System.Web.UI.Design.ITemplateEditingFrame CreateTemplateEditingFrame(
   System.Web.UI.Design.TemplateEditingVerb verb
);
protected abstract System.Web.UI.Design.TemplateEditingVerb[] GetCachedTemplateEditingVerbs()
;
protected System.Web.UI.ITemplate GetTemplateFromText(
   System.String text
);
protected System.String GetTextFromTemplate(
   System.Web.UI.ITemplate template
);
protected void OnBehaviorAttached();
protected void OnTemplateModeChanged();
protected void PreFilterProperties(
   System.Collections.IDictionary properties
);
protected void SaveActiveTemplateEditingFrame();
```

TemplateEditingService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TemplateEditingService
   extends System.Object
   implements System.Web.UI.Design.ITemplateEditingService, System.IDisposable
```

Public Constructors

```
public TemplateEditingService(
   System.ComponentModel.Design.IDesignerHost designerHost
);
```

Public Properties

```
public final boolean get_SupportsNestedTemplateEditing();
```

Public Methods

```
public final System.Web.UI.Design.ITemplateEditingFrame CreateFrame(
   System.Web.UI.Design.TemplatedControlDesigner designer,
   System.String frameName,
   System.String[] templateNames
);
public final System.Web.UI.Design.ITemplateEditingFrame CreateFrame(
   System.Web.UI.Design.TemplatedControlDesigner designer,
   System.String frameName,
   System.String[] templateNames,
   System.Web.UI.WebControls.Style controlStyle,
   System.Web.UI.WebControls.Style[] templateStyles
);
public final void Dispose();
public final System.String GetContainingTemplateName(
   System.Web.UI.Control control
);
```

Protected Methods

```
protected void Finalize();
```

TemplateEditingVerb Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TemplateEditingVerb
   extends System.ComponentModel.Design.DesignerVerb
   implements System.IDisposable
```

Public Constructors

```
public TemplateEditingVerb(
   System.String text,
   int index,
   System.Web.UI.Design.TemplatedControlDesigner designer
);
```

Public Properties

```
public int get_Index();
```

Public Methods

```
public final void Dispose();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

TextControlDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TextControlDesigner
   extends System.Web.UI.Design.ControlDesigner
```

Public Constructors

```
public TextControlDesigner();
```

Public Methods

```
public System.String GetDesignTimeHtml();
public System.String GetPersistInnerHtml();
public void Initialize(
   System.ComponentModel.IComponent component
);
```

TextDataBindingHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TextDataBindingHandler
   extends System.Web.UI.Design.DataBindingHandler
```

Public Constructors

```
public TextDataBindingHandler();
```

Public Methods

```
public void DataBindControl(
   System.ComponentModel.Design.IDesignerHost designerHost,
   System.Web.UI.Control control
);
```

UrlBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UrlBuilder
   extends System.Object
```

Public Methods

```
public static System.String BuildUrl(
   System.ComponentModel.IComponent component,
   System.Windows.Forms.Control owner,
   System.String initialUrl,
   System.String caption,
   System.String filter
);
public static System.String BuildUrl(
   System.ComponentModel.IComponent component,
   System.Windows.Forms.Control owner,
   System.String initialUrl,
   System.String caption,
   System.String filter,
   System.Web.UI.Design.UrlBuilderOptions options
);
```

UrlBuilderOptions Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum UrlBuilderOptions
```

Members

NoAbsolute

None

UrlEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UrlEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public UrlEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

Protected Properties

```
protected System.String get_Caption();
protected System.String get_Filter();
protected System.Web.UI.Design.UrlBuilderOptions get_Options();
```

UserControlDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UserControlDesigner
   extends System.Web.UI.Design.ControlDesigner
```

Public Constructors

```
public UserControlDesigner();
```

Public Properties

```
public boolean get_AllowResize();
public boolean get_ShouldCodeSerialize();
public void set_ShouldCodeSerialize(
   boolean value
);
```

Public Methods

```
public System.String GetDesignTimeHtml();
public System.String GetPersistInnerHtml();
```

WebControlToolboxItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class WebControlToolboxItem
   extends System.Drawing.Design.ToolboxItem
```

Public Constructors

```
public WebControlToolboxItem();
public WebControlToolboxItem(
   System.Type type
);
```

Public Methods

```
public System.Object GetToolAttributeValue(
   System.ComponentModel.Design.IDesignerHost host,
   System.Type attributeType
);
public System.String GetToolHtml(
   System.ComponentModel.Design.IDesignerHost host
);
public System.Type GetToolType(
   System.ComponentModel.Design.IDesignerHost host
);
public void Initialize(
   System.Type type
);
```

Protected Methods

```
protected System.ComponentModel.IComponent[] CreateComponentsCore(
   System.ComponentModel.Design.IDesignerHost host
);
protected void Deserialize(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
protected void Serialize(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

XmlFileEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlFileEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public XmlFileEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

XmlUrlEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlUrlEditor
   extends System.Web.UI.Design.UrlEditor
```

Public Constructors

```
public XmlUrlEditor();
```

Protected Properties

```
protected System.String get_Caption();
protected System.String get_Filter();
protected System.Web.UI.Design.UrlBuilderOptions get_Options();
```

XslUrlEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XslUrlEditor
   extends System.Web.UI.Design.UrlEditor
```

Public Constructors

```
public XslUrlEditor();
```

Protected Properties

```
protected System.String get_Caption();
protected System.String get_Filter();
protected System.Web.UI.Design.UrlBuilderOptions get_Options();
```

System.Web.UI.Design.WebControls Namespace (Visual J# Syntax)

Classes

AdRotatorDesigner

BaseDataListComponentEditor

BaseDataListDesigner

BaseValidatorDesigner

ButtonDesigner

CalendarAutoFormatDialog

CalendarDesigner

CheckBoxDesigner

DataGridColumnCollectionEditor

DataGridComponentEditor

DataGridDesigner

DataListComponentEditor

DataListDesigner

HyperLinkDesigner

LabelDesigner

LinkButtonDesigner

ListControlDataBindingHandler

ListControlDesigner

ListItemsCollectionEditor

PanelDesigner

RegexEditorDialog

RegexTypeEditor

RepeaterDesigner

TableCellsCollectionEditor

TableDesigner

TableRowsCollectionEditor

XmlDesigner

AdRotatorDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AdRotatorDesigner
   extends System.Web.UI.Design.ControlDesigner
```

Public Constructors

```
public AdRotatorDesigner();
```

Public Methods

```
public System.String GetDesignTimeHtml();
```

BaseDataListComponentEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class BaseDataListComponentEditor
   extends System.Windows.Forms.Design.WindowsFormsComponentEditor
```

Public Constructors

```
public BaseDataListComponentEditor(
   int initialPage
);
```

Public Methods

```
public boolean EditComponent(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object obj,
   System.Windows.Forms.IWin32Window parent
);
```

Protected Methods

```
protected int GetInitialComponentEditorPageIndex();
```

BaseDataListDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class BaseDataListDesigner
   extends System.Web.UI.Design.TemplatedControlDesigner
   implements System.Web.UI.Design.IDataSourceProvider
```

Public Constructors

```
public BaseDataListDesigner();
```

Public Properties

```
public System.String get_DataKeyField();
public void set_DataKeyField(
   System.String value
);
public System.String get_DataMember();
public void set_DataMember(
   System.String value
);
public System.String get_DataSource();
public void set_DataSource(
   System.String value
);
public boolean get_DesignTimeHtmlRequiresLoadComplete();
public System.ComponentModel.Design.DesignerVerbCollection get_Verbs();
```

Public Methods

```
public final System.Collections.IEnumerable GetResolvedSelectedDataSource();
public final System.Object GetSelectedDataSource();
public System.Collections.IEnumerable GetTemplateContainerDataSource(
   System.String templateName
);
public void Initialize(
   System.ComponentModel.IComponent component
);
public void OnComponentChanged(
   System.Object sender,
   System.ComponentModel.Design.ComponentChangedEventArgs e
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected System.Collections.IEnumerable GetDesignTimeDataSource(
   int minimumRows,
   boolean dummyDataSource
);
protected System.Collections.IEnumerable GetDesignTimeDataSource(
   System.Collections.IEnumerable selectedDataSource,
```

```
   int minimumRows,
   boolean dummyDataSource
);
protected void InvokePropertyBuilder(
   int initialPage
);
protected void OnAutoFormat(
   System.Object sender,
   System.EventArgs e
);
protected void OnDataSourceChanged();
protected void OnPropertyBuilder(
   System.Object sender,
   System.EventArgs e
);
protected void OnStylesChanged();
protected abstract void OnTemplateEditingVerbsChanged();
protected void PreFilterProperties(
   System.Collections.IDictionary properties
);
```

BaseValidatorDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BaseValidatorDesigner
   extends System.Web.UI.Design.ControlDesigner
```

Public Constructors

```
public BaseValidatorDesigner();
```

Public Methods

```
public System.String GetDesignTimeHtml();
```

ButtonDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ButtonDesigner
   extends System.Web.UI.Design.ControlDesigner
```

Public Constructors

```
public ButtonDesigner();
```

Public Methods

```
public System.String GetDesignTimeHtml();
```

CalendarAutoFormatDialog Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class CalendarAutoFormatDialog
   extends System.Windows.Forms.Form
```

Public Constructors

```
public CalendarAutoFormatDialog(
   System.Web.UI.WebControls.Calendar calendar
);
```

Protected Methods

```
protected void DoDelayLoadActions();
protected void OnActivated(
   System.Object source,
   System.EventArgs e
);
protected void OnOKClicked(
   System.Object source,
   System.EventArgs e
);
protected void OnSelChangedScheme(
   System.Object source,
   System.EventArgs e
);
protected void SaveComponent();
```

CalendarDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CalendarDesigner
   extends System.Web.UI.Design.ControlDesigner
```

Public Constructors

```
public CalendarDesigner();
```

Public Properties

```
public System.ComponentModel.Design.DesignerVerbCollection get_Verbs();
```

Public Methods

```
public void Initialize(
   System.ComponentModel.IComponent component
);
```

Protected Methods

```
protected void OnAutoFormat(
   System.Object sender,
   System.EventArgs e
);
```

CheckBoxDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CheckBoxDesigner
   extends System.Web.UI.Design.ControlDesigner
```

Public Constructors

```
public CheckBoxDesigner();
```

Public Methods

```
public System.String GetDesignTimeHtml();
```

DataGridColumnCollectionEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataGridColumnCollectionEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public DataGridColumnCollectionEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

DataGridComponentEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataGridComponentEditor
   extends System.Web.UI.Design.WebControls.BaseDataListComponentEditor
```

Public Constructors

```
public DataGridComponentEditor();
public DataGridComponentEditor(
   int initialPage
);
```

Protected Methods

```
protected System.Type[] GetComponentEditorPages();
```

DataGridDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataGridDesigner
   extends System.Web.UI.Design.WebControls.BaseDataListDesigner
```

Public Constructors

```
public DataGridDesigner();
```

Public Methods

```
public System.String GetDesignTimeHtml();
public System.String GetTemplateContainerDataItemProperty(
   System.String templateName
);
public System.String GetTemplateContent(
   System.Web.UI.Design.ITemplateEditingFrame editingFrame,
   System.String templateName,
   boolean allowEditing
);
public System.Type GetTemplatePropertyParentType(
   System.String templateName
);
public void Initialize(
   System.ComponentModel.IComponent component
);
public void OnColumnsChanged();
public void SetTemplateContent(
   System.Web.UI.Design.ITemplateEditingFrame editingFrame,
   System.String templateName,
   System.String templateContent
);
```

Protected Methods

```
protected System.Web.UI.Design.ITemplateEditingFrame CreateTemplateEditingFrame(
   System.Web.UI.Design.TemplateEditingVerb verb
);
protected void Dispose(
   boolean disposing
);
protected System.Web.UI.Design.TemplateEditingVerb[] GetCachedTemplateEditingVerbs();
protected System.String GetEmptyDesignTimeHtml();
protected System.String GetErrorDesignTimeHtml(
   System.Exception e
);
protected void OnTemplateEditingVerbsChanged();
```

DataListComponentEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataListComponentEditor
   extends System.Web.UI.Design.WebControls.BaseDataListComponentEditor
```

Public Constructors

```
public DataListComponentEditor();
public DataListComponentEditor(
   int initialPage
);
```

Protected Methods

```
protected System.Type[] GetComponentEditorPages();
```

DataListDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataListDesigner
   extends System.Web.UI.Design.WebControls.BaseDataListDesigner
```

Public Constructors

```
public DataListDesigner();
```

Public Properties

```
public boolean get_AllowResize();
```

Public Methods

```
public System.String GetDesignTimeHtml();
public System.String GetTemplateContainerDataItemProperty(
   System.String templateName
);
public System.String GetTemplateContent(
   System.Web.UI.Design.ITemplateEditingFrame editingFrame,
   System.String templateName,
   boolean allowEditing
);
public void Initialize(
   System.ComponentModel.IComponent component
);
public void SetTemplateContent(
   System.Web.UI.Design.ITemplateEditingFrame editingFrame,
   System.String templateName,
   System.String templateContent
);
```

Protected Properties

```
protected boolean get_TemplatesExist();
```

Protected Methods

```
protected System.Web.UI.Design.ITemplateEditingFrame CreateTemplateEditingFrame(
   System.Web.UI.Design.TemplateEditingVerb verb
);
protected void Dispose(
   boolean disposing
);
protected System.Web.UI.Design.TemplateEditingVerb[] GetCachedTemplateEditingVerbs();
protected System.String GetEmptyDesignTimeHtml();
protected System.String GetErrorDesignTimeHtml(
   System.Exception e
);
protected void OnTemplateEditingVerbsChanged();
```

HyperLinkDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HyperLinkDesigner
   extends System.Web.UI.Design.TextControlDesigner
```

Public Constructors

```
public HyperLinkDesigner();
```

Public Methods

```
public System.String GetDesignTimeHtml();
```

LabelDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LabelDesigner
   extends System.Web.UI.Design.TextControlDesigner
```

Public Constructors

```
public LabelDesigner();
```

LinkButtonDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LinkButtonDesigner
   extends System.Web.UI.Design.TextControlDesigner
```

Public Constructors

```
public LinkButtonDesigner();
```

ListControlDataBindingHandler Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ListControlDataBindingHandler
   extends System.Web.UI.Design.DataBindingHandler
```

Public Constructors

```
public ListControlDataBindingHandler();
```

Public Methods

```
public void DataBindControl(
   System.ComponentModel.Design.IDesignerHost designerHost,
   System.Web.UI.Control control
);
```

ListControlDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ListControlDesigner
   extends System.Web.UI.Design.ControlDesigner
   implements System.Web.UI.Design.IDataSourceProvider
```

Public Constructors

```
public ListControlDesigner();
```

Public Properties

```
public System.String get_DataMember();
public void set_DataMember(
   System.String value
);
public System.String get_DataSource();
public void set_DataSource(
   System.String value
);
public System.String get_DataTextField();
public void set_DataTextField(
   System.String value
);
public System.String get_DataValueField();
public void set_DataValueField(
   System.String value
);
```

Public Methods

```
public System.String GetDesignTimeHtml();
public final System.Collections.IEnumerable GetResolvedSelectedDataSource();
public final System.Object GetSelectedDataSource();
public void Initialize(
   System.ComponentModel.IComponent component
);
public void OnComponentChanged(
   System.Object source,
   System.ComponentModel.Design.ComponentChangedEventArgs ce
);
public void OnDataSourceChanged();
```

Protected Methods

```
protected void PreFilterProperties(
   System.Collections.IDictionary properties
);
```

ListItemsCollectionEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ListItemsCollectionEditor
   extends System.ComponentModel.Design.CollectionEditor
```

Public Constructors

```
public ListItemsCollectionEditor(
   System.Type type
);
```

Protected Methods

```
protected boolean CanSelectMultipleInstances();
```

PanelDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PanelDesigner
   extends System.Web.UI.Design.ReadWriteControlDesigner
```

Public Constructors

```
public PanelDesigner();
```

Protected Methods

```
protected void MapPropertyToStyle(
   System.String propName,
   System.Object varPropValue
);
protected void OnBehaviorAttached();
```

RegexEditorDialog Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class RegexEditorDialog
   extends System.Windows.Forms.Form
```

Public Constructors

```
public RegexEditorDialog(
   System.ComponentModel.ISite site
);
```

Public Properties

```
public System.String get_RegularExpression();
public void set_RegularExpression(
   System.String value
);
```

Protected Methods

```
protected void cmdHelp_Click(
   System.Object sender,
   System.EventArgs e
);
protected void cmdOK_Click(
   System.Object sender,
   System.EventArgs e
);
protected void cmdTestValidate_Click(
   System.Object sender,
   System.EventArgs args
);
protected void Dispose(
   boolean disposing
);
protected void lstStandardExpressions_SelectedIndexChanged(
   System.Object sender,
   System.EventArgs e
);
protected void RegexTypeEditor_Activated(
   System.Object sender,
   System.EventArgs e
);
protected void txtExpression_TextChanged(
   System.Object sender,
   System.EventArgs e
);
```

RegexTypeEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RegexTypeEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public RegexTypeEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

RepeaterDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RepeaterDesigner
   extends System.Web.UI.Design.ControlDesigner
   implements System.Web.UI.Design.IDataSourceProvider
```

Public Constructors

```
public RepeaterDesigner();
```

Public Properties

```
public System.String get_DataMember();
public void set_DataMember(
   System.String value
);
public System.String get_DataSource();
public void set_DataSource(
   System.String value
);
```

Public Methods

```
public System.String GetDesignTimeHtml();
public final System.Collections.IEnumerable GetResolvedSelectedDataSource();
public final System.Object GetSelectedDataSource();
public void Initialize(
   System.ComponentModel.IComponent component
);
public void OnComponentChanged(
   System.Object source,
   System.ComponentModel.Design.ComponentChangedEventArgs ce
);
public void OnDataSourceChanged();
```

Protected Properties

```
protected boolean get_TemplatesExist();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected System.Collections.IEnumerable GetDesignTimeDataSource(
   int minimumRows
);
protected System.Collections.IEnumerable GetDesignTimeDataSource(
   System.Collections.IEnumerable selectedDataSource,
   int minimumRows
);
```

```
protected System.String GetEmptyDesignTimeHtml();
protected System.String GetErrorDesignTimeHtml(
   System.Exception e
);
protected void PreFilterProperties(
   System.Collections.IDictionary properties
);
```

TableCellsCollectionEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TableCellsCollectionEditor
   extends System.ComponentModel.Design.CollectionEditor
```

Public Constructors

```
public TableCellsCollectionEditor(
   System.Type type
);
```

Protected Methods

```
protected boolean CanSelectMultipleInstances();
protected System.Object CreateInstance(
   System.Type itemType
);
```

TableDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TableDesigner
   extends System.Web.UI.Design.ControlDesigner
```

Public Constructors

```
public TableDesigner();
```

Public Methods

```
public System.String GetDesignTimeHtml();
public System.String GetPersistInnerHtml();
```

TableRowsCollectionEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TableRowsCollectionEditor
   extends System.ComponentModel.Design.CollectionEditor
```

Public Constructors

```
public TableRowsCollectionEditor(
   System.Type type
);
```

Protected Methods

```
protected boolean CanSelectMultipleInstances();
protected System.Object CreateInstance(
   System.Type itemType
);
```

XmlDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlDesigner
   extends System.Web.UI.Design.ControlDesigner
```

Public Constructors

```
public XmlDesigner();
```

Public Methods

```
public System.String GetDesignTimeHtml();
public void Initialize(
   System.ComponentModel.IComponent component
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected System.String GetEmptyDesignTimeHtml();
```

System.Web.UI.HtmlControls Namespace (Visual J# Syntax)

Classes

HtmlAnchor

HtmlButton

HtmlContainerControl

HtmlControl

HtmlForm

HtmlGenericControl

HtmlImage

HtmlInputButton

HtmlInputCheckBox

HtmlInputControl

HtmlInputFile

HtmlInputHidden

HtmlInputImage

HtmlInputRadioButton

HtmlInputText

HtmlSelect

HtmlTable

HtmlTable.HtmlTableRowControlCollection

HtmlTableCell

HtmlTableCellCollection

HtmlTableRow

HtmlTableRow.HtmlTableCellControlCollection

HtmlTableRowCollection

HtmlTextArea

HtmlAnchor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ServerClick") */
public class HtmlAnchor
   extends System.Web.UI.HtmlControls.HtmlContainerControl
   implements System.Web.UI.IPostBackEventHandler
```

Public Constructors

```
public HtmlAnchor();
```

Public Properties

```
public System.String get_HRef();
public void set_HRef(
   System.String value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_Target();
public void set_Target(
   System.String value
);
public System.String get_Title();
public void set_Title(
   System.String value
);
```

Public Events

```
public void add_ServerClick(
   System.EventHandler value
);
public void remove_ServerClick(
   System.EventHandler value
);
```

Protected Methods

```
protected void OnPreRender(
   System.EventArgs e
);
protected void OnServerClick(
   System.EventArgs e
);
protected void RenderAttributes(
   System.Web.UI.HtmlTextWriter writer
);
```

HtmlButton Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ServerClick") */
public class HtmlButton
   extends System.Web.UI.HtmlControls.HtmlContainerControl
   implements System.Web.UI.IPostBackEventHandler
```

Public Constructors

```
public HtmlButton();
```

Public Properties

```
public boolean get_CausesValidation();
public void set_CausesValidation(
   boolean value
);
```

Public Events

```
public void add_ServerClick(
   System.EventHandler value
);
public void remove_ServerClick(
   System.EventHandler value
);
```

Protected Methods

```
protected void OnPreRender(
   System.EventArgs e
);
protected void OnServerClick(
   System.EventArgs e
);
protected void RenderAttributes(
   System.Web.UI.HtmlTextWriter writer
);
```

HtmlContainerControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class HtmlContainerControl
   extends System.Web.UI.HtmlControls.HtmlControl
```

Public Constructors

```
public HtmlContainerControl();
public HtmlContainerControl(
   System.String tag
);
```

Public Properties

```
public System.String get_InnerHtml();
public void set_InnerHtml(
   System.String value
);
public System.String get_InnerText();
public void set_InnerText(
   System.String value
);
```

Protected Methods

```
protected System.Web.UI.ControlCollection CreateControlCollection();
protected void LoadViewState(
   System.Object savedState
);
protected void Render(
   System.Web.UI.HtmlTextWriter writer
);
protected void RenderAttributes(
   System.Web.UI.HtmlTextWriter writer
);
protected void RenderEndTag(
   System.Web.UI.HtmlTextWriter writer
);
```

HtmlControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.HtmlIntrinsicControlDesig
ner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public abstract class HtmlControl
   extends System.Web.UI.Control
   implements System.Web.UI.IAttributeAccessor
```

Public Constructors

```
public HtmlControl();
public HtmlControl(
   System.String tag
);
```

Public Properties

```
public System.Web.UI.AttributeCollection get_Attributes();
public boolean get_Disabled();
public void set_Disabled(
   boolean value
);
public System.Web.UI.CssStyleCollection get_Style();
public System.String get_TagName();
```

Protected Properties

```
protected boolean get_ViewStateIgnoresCase();
```

Protected Methods

```
protected System.Web.UI.ControlCollection CreateControlCollection();
protected void Render(
   System.Web.UI.HtmlTextWriter writer
);
protected void RenderAttributes(
   System.Web.UI.HtmlTextWriter writer
);
protected void RenderBeginTag(
   System.Web.UI.HtmlTextWriter writer
);
```

HtmlForm Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HtmlForm
   extends System.Web.UI.HtmlControls.HtmlContainerControl
```

Public Constructors

```
public HtmlForm();
```

Public Properties

```
public System.String get_Enctype();
public void set_Enctype(
   System.String value
);
public System.String get_Method();
public void set_Method(
   System.String value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_Target();
public void set_Target(
   System.String value
);
```

Protected Methods

```
protected void OnInit(
   System.EventArgs e
);
protected void Render(
   System.Web.UI.HtmlTextWriter output
);
protected void RenderAttributes(
   System.Web.UI.HtmlTextWriter writer
);
protected void RenderChildren(
   System.Web.UI.HtmlTextWriter writer
);
```

HtmlGenericControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ConstructorNeedsTagAttribute(NeedsTag = true) */
public class HtmlGenericControl
   extends System.Web.UI.HtmlControls.HtmlContainerControl
```

Public Constructors

```
public HtmlGenericControl();
public HtmlGenericControl(
   System.String tag
);
```

Public Properties

```
public System.String get_TagName();
public void set_TagName(
   System.String value
);
```

HtmlImage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ControlBuilderAttribute(BuilderType = System.Web.UI.HtmlControls
.HtmlControlBuilder) */
public class HtmlImage
   extends System.Web.UI.HtmlControls.HtmlControl
```

Public Constructors

```
public HtmlImage();
```

Public Properties

```
public System.String get_Align();
public void set_Align(
   System.String value
);
public System.String get_Alt();
public void set_Alt(
   System.String value
);
public int get_Border();
public void set_Border(
   int value
);
public int get_Height();
public void set_Height(
   int value
);
public System.String get_Src();
public void set_Src(
   System.String value
);
public int get_Width();
public void set_Width(
   int value
);
```

Protected Methods

```
protected void RenderAttributes(
   System.Web.UI.HtmlTextWriter writer
);
```

HtmlInputButton Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ServerClick") */
public class HtmlInputButton
   extends System.Web.UI.HtmlControls.HtmlInputControl
   implements System.Web.UI.IPostBackEventHandler
```

Public Constructors

```
public HtmlInputButton();
public HtmlInputButton(
   System.String type
);
```

Public Properties

```
public boolean get_CausesValidation();
public void set_CausesValidation(
   boolean value
);
```

Public Events

```
public void add_ServerClick(
   System.EventHandler value
);
public void remove_ServerClick(
   System.EventHandler value
);
```

Protected Methods

```
protected void OnPreRender(
   System.EventArgs e
);
protected void OnServerClick(
   System.EventArgs e
);
protected void RenderAttributes(
   System.Web.UI.HtmlTextWriter writer
);
```

HtmlInputCheckBox Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ServerChange") */
public class HtmlInputCheckBox
   extends System.Web.UI.HtmlControls.HtmlInputControl
   implements System.Web.UI.IPostBackDataHandler
```

Public Constructors

```
public HtmlInputCheckBox();
```

Public Properties

```
public boolean get_Checked();
public void set_Checked(
   boolean value
);
```

Public Events

```
public void add_ServerChange(
   System.EventHandler value
);
public void remove_ServerChange(
   System.EventHandler value
);
```

Protected Methods

```
protected void OnPreRender(
   System.EventArgs e
);
protected void OnServerChange(
   System.EventArgs e
);
```

HtmlInputControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ControlBuilderAttribute(BuilderType = System.Web.UI.HtmlControls
.HtmlControlBuilder) */
public abstract class HtmlInputControl
   extends System.Web.UI.HtmlControls.HtmlControl
```

Public Constructors

```
public HtmlInputControl(
   System.String type
);
```

Public Properties

```
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_Type();
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Protected Methods

```
protected void RenderAttributes(
   System.Web.UI.HtmlTextWriter writer
);
```

HtmlInputFile Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ValidationPropertyAttribute(Name = "Value") */
public class HtmlInputFile
   extends System.Web.UI.HtmlControls.HtmlInputControl
   implements System.Web.UI.IPostBackDataHandler
```

Public Constructors

```
public HtmlInputFile();
```

Public Properties

```
public System.String get_Accept();
public void set_Accept(
   System.String value
);
public int get_MaxLength();
public void set_MaxLength(
   int value
);
public System.Web.HttpPostedFile get_PostedFile();
public int get_Size();
public void set_Size(
   int value
);
```

HtmlInputHidden Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ServerChange") */
public class HtmlInputHidden
   extends System.Web.UI.HtmlControls.HtmlInputControl
   implements System.Web.UI.IPostBackDataHandler
```

Public Constructors

```
public HtmlInputHidden();
```

Public Events

```
public void add_ServerChange(
   System.EventHandler value
);
public void remove_ServerChange(
   System.EventHandler value
);
```

Protected Methods

```
protected void OnPreRender(
   System.EventArgs e
);
protected void OnServerChange(
   System.EventArgs e
);
```

HtmlInputImage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ServerClick") */
public class HtmlInputImage
   extends System.Web.UI.HtmlControls.HtmlInputControl
   implements System.Web.UI.IPostBackDataHandler, System.Web.UI.IPostBackEventHandler
```

Public Constructors

```
public HtmlInputImage();
```

Public Properties

```
public System.String get_Align();
public void set_Align(
   System.String value
);
public System.String get_Alt();
public void set_Alt(
   System.String value
);
public int get_Border();
public void set_Border(
   int value
);
public boolean get_CausesValidation();
public void set_CausesValidation(
   boolean value
);
public System.String get_Src();
public void set_Src(
   System.String value
);
```

Public Events

```
public void add_ServerClick(
   System.Web.UI.ImageClickEventHandler value
);
public void remove_ServerClick(
   System.Web.UI.ImageClickEventHandler value
);
```

Protected Methods

```
protected void OnPreRender(
   System.EventArgs e
);
protected void OnServerClick(
   System.Web.UI.ImageClickEventArgs e
);
protected void RenderAttributes(
   System.Web.UI.HtmlTextWriter writer
```

```
);
```

HtmlInputRadioButton Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ServerChange") */
public class HtmlInputRadioButton
   extends System.Web.UI.HtmlControls.HtmlInputControl
   implements System.Web.UI.IPostBackDataHandler
```

Public Constructors

```
public HtmlInputRadioButton();
```

Public Properties

```
public boolean get_Checked();
public void set_Checked(
   boolean value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Events

```
public void add_ServerChange(
   System.EventHandler value
);
public void remove_ServerChange(
   System.EventHandler value
);
```

Protected Methods

```
protected void OnPreRender(
   System.EventArgs e
);
protected void OnServerChange(
   System.EventArgs e
);
protected void RenderAttributes(
   System.Web.UI.HtmlTextWriter writer
);
```

HtmlInputText Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ValidationPropertyAttribute(Name = "Value") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ServerChange") */
public class HtmlInputText
   extends System.Web.UI.HtmlControls.HtmlInputControl
   implements System.Web.UI.IPostBackDataHandler
```

Public Constructors

```
public HtmlInputText();
public HtmlInputText(
   System.String type
);
```

Public Properties

```
public int get_MaxLength();
public void set_MaxLength(
   int value
);
public int get_Size();
public void set_Size(
   int value
);
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Events

```
public void add_ServerChange(
   System.EventHandler value
);
public void remove_ServerChange(
   System.EventHandler value
);
```

Protected Methods

```
protected void OnPreRender(
   System.EventArgs e
);
protected void OnServerChange(
   System.EventArgs e
);
protected void RenderAttributes(
   System.Web.UI.HtmlTextWriter writer
);
```

HtmlSelect Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ControlBuilderAttribute(BuilderType = System.Web.UI.HtmlControls
.HtmlSelectBuilder) */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ServerChange") */
/** @attribute System.Web.UI.ValidationPropertyAttribute(Name = "Value") */
public class HtmlSelect
   extends System.Web.UI.HtmlControls.HtmlContainerControl
   implements System.Web.UI.IPostBackDataHandler
```

Public Constructors

```
public HtmlSelect();
```

Public Properties

```
public System.String get_DataMember();
public void set_DataMember(
   System.String value
);
public System.Object get_DataSource();
public void set_DataSource(
   System.Object value
);
public System.String get_DataTextField();
public void set_DataTextField(
   System.String value
);
public System.String get_DataValueField();
public void set_DataValueField(
   System.String value
);
public System.String get_InnerHtml();
public void set_InnerHtml(
   System.String value
);
public System.String get_InnerText();
public void set_InnerText(
   System.String value
);
public System.Web.UI.WebControls.ListItemCollection get_Items();
public boolean get_Multiple();
public void set_Multiple(
   boolean value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public int get_SelectedIndex();
public void set_SelectedIndex(
   int value
);
public int get_Size();
public void set_Size(
   int value
);
```

```
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Events

```
public void add_ServerChange(
   System.EventHandler value
);
public void remove_ServerChange(
   System.EventHandler value
);
```

Protected Properties

```
protected int[] get_SelectedIndices();
```

Protected Methods

```
protected void AddParsedSubObject(
   System.Object obj
);
protected void ClearSelection();
protected System.Web.UI.ControlCollection CreateControlCollection();
protected void LoadViewState(
   System.Object savedState
);
protected void OnDataBinding(
   System.EventArgs e
);
protected void OnPreRender(
   System.EventArgs e
);
protected void OnServerChange(
   System.EventArgs e
);
protected void RenderAttributes(
   System.Web.UI.HtmlTextWriter writer
);
protected void RenderChildren(
   System.Web.UI.HtmlTextWriter writer
);
protected System.Object SaveViewState();
protected void Select(
   int[] selectedIndices
);
protected void TrackViewState();
```

HtmlTable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ParseChildrenAttribute(ChildrenAsProperties = true, DefaultProperty = "Rows") */
public class HtmlTable
   extends System.Web.UI.HtmlControls.HtmlContainerControl
```

Public Constructors

```
public HtmlTable();
```

Public Properties

```
public System.String get_Align();
public void set_Align(
   System.String value
);
public System.String get_BgColor();
public void set_BgColor(
   System.String value
);
public int get_Border();
public void set_Border(
   int value
);
public System.String get_BorderColor();
public void set_BorderColor(
   System.String value
);
public int get_CellPadding();
public void set_CellPadding(
   int value
);
public int get_CellSpacing();
public void set_CellSpacing(
   int value
);
public System.String get_Height();
public void set_Height(
   System.String value
);
public System.String get_InnerHtml();
public void set_InnerHtml(
   System.String value
);
public System.String get_InnerText();
public void set_InnerText(
   System.String value
);
public System.Web.UI.HtmlControls.HtmlTableRowCollection get_Rows();
public System.String get_Width();
public void set_Width(
   System.String value
);
```

Protected Methods

```
protected System.Web.UI.ControlCollection CreateControlCollection();
protected void RenderChildren(
   System.Web.UI.HtmlTextWriter writer
);
protected void RenderEndTag(
   System.Web.UI.HtmlTextWriter writer
);
```

HtmlTable.HtmlTableRowControlCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
protected class HtmlTableRowControlCollection
   extends System.Web.UI.ControlCollection
```

Public Methods

```
public void Add(
   System.Web.UI.Control child
);
public void AddAt(
   int index,
   System.Web.UI.Control child
);
```

HtmlTableCell Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ConstructorNeedsTagAttribute(NeedsTag = true) */
public class HtmlTableCell
   extends System.Web.UI.HtmlControls.HtmlContainerControl
```

Public Constructors

```
public HtmlTableCell();
public HtmlTableCell(
   System.String tagName
);
```

Public Properties

```
public System.String get_Align();
public void set_Align(
   System.String value
);
public System.String get_BgColor();
public void set_BgColor(
   System.String value
);
public System.String get_BorderColor();
public void set_BorderColor(
   System.String value
);
public int get_ColSpan();
public void set_ColSpan(
   int value
);
public System.String get_Height();
public void set_Height(
   System.String value
);
public boolean get_NoWrap();
public void set_NoWrap(
   boolean value
);
public int get_RowSpan();
public void set_RowSpan(
   int value
);
public System.String get_VAlign();
public void set_VAlign(
   System.String value
);
public System.String get_Width();
public void set_Width(
   System.String value
);
```

Protected Methods

```
protected void RenderEndTag(
```

```
    System.Web.UI.HtmlTextWriter writer
);
```

HtmlTableCellCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HtmlTableCellCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Web.UI.HtmlControls.HtmlTableCell get_Item(
   int index
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public void Add(
   System.Web.UI.HtmlControls.HtmlTableCell cell
);
public void Clear();
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public void Insert(
   int index,
   System.Web.UI.HtmlControls.HtmlTableCell cell
);
public void Remove(
   System.Web.UI.HtmlControls.HtmlTableCell cell
);
public void RemoveAt(
   int index
);
```

HtmlTableRow Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ParseChildrenAttribute(ChildrenAsProperties = true, DefaultProperty = "Cells") */
public class HtmlTableRow
   extends System.Web.UI.HtmlControls.HtmlContainerControl
```

Public Constructors

```
public HtmlTableRow();
```

Public Properties

```
public System.String get_Align();
public void set_Align(
   System.String value
);
public System.String get_BgColor();
public void set_BgColor(
   System.String value
);
public System.String get_BorderColor();
public void set_BorderColor(
   System.String value
);
public System.Web.UI.HtmlControls.HtmlTableCellCollection get_Cells();
public System.String get_Height();
public void set_Height(
   System.String value
);
public System.String get_InnerHtml();
public void set_InnerHtml(
   System.String value
);
public System.String get_InnerText();
public void set_InnerText(
   System.String value
);
public System.String get_VAlign();
public void set_VAlign(
   System.String value
);
```

Protected Methods

```
protected System.Web.UI.ControlCollection CreateControlCollection();
protected void RenderChildren(
   System.Web.UI.HtmlTextWriter writer
);
protected void RenderEndTag(
   System.Web.UI.HtmlTextWriter writer
);
```

HtmlTableRow.HtmlTableCellControlCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
protected class HtmlTableCellControlCollection
   extends System.Web.UI.ControlCollection
```

Public Methods

```
public void Add(
   System.Web.UI.Control child
);
public void AddAt(
   int index,
   System.Web.UI.Control child
);
```

HtmlTableRowCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class HtmlTableRowCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Web.UI.HtmlControls.HtmlTableRow get_Item(
   int index
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public void Add(
   System.Web.UI.HtmlControls.HtmlTableRow row
);
public void Clear();
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public void Insert(
   int index,
   System.Web.UI.HtmlControls.HtmlTableRow row
);
public void Remove(
   System.Web.UI.HtmlControls.HtmlTableRow row
);
public void RemoveAt(
   int index
);
```

HtmlTextArea Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ServerChange") */
/** @attribute System.Web.UI.ValidationPropertyAttribute(Name = "Value") */
public class HtmlTextArea
   extends System.Web.UI.HtmlControls.HtmlContainerControl
   implements System.Web.UI.IPostBackDataHandler
```

Public Constructors

```
public HtmlTextArea();
```

Public Properties

```
public int get_Cols();
public void set_Cols(
   int value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public int get_Rows();
public void set_Rows(
   int value
);
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Events

```
public void add_ServerChange(
   System.EventHandler value
);
public void remove_ServerChange(
   System.EventHandler value
);
```

Protected Methods

```
protected void AddParsedSubObject(
   System.Object obj
);
protected void OnPreRender(
   System.EventArgs e
);
protected void OnServerChange(
   System.EventArgs e
);
protected void RenderAttributes(
   System.Web.UI.HtmlTextWriter writer
```

```
);
```

System.Web.UI.WebControls Namespace (Visual J# Syntax)

Classes

AdCreatedEventArgs

AdRotator

BaseCompareValidator

BaseDataList

BaseValidator

BoundColumn

Button

ButtonColumn

Calendar

CalendarDay

CheckBox

CheckBoxList

CommandEventArgs

CompareValidator

CustomValidator

DataGrid

DataGridColumn

DataGridColumnCollection

DataGridCommandEventArgs

DataGridItem

DataGridItemCollection

DataGridItemEventArgs

DataGridPageChangedEventArgs

DataGridPagerStyle

DataGridSortCommandEventArgs

DataKeyCollection

DataList

DataListCommandEventArgs

DataListItem

DataListItemCollection

DataListItemEventArgs

DayRenderEventArgs

DropDownList

EditCommandColumn

FontInfo

FontNamesConverter

FontUnitConverter

HyperLink

HyperLinkColumn

HyperLinkControlBuilder

Image

ImageButton

Label

LabelControlBuilder

LinkButton

LinkButtonControlBuilder

ListBox

ListControl

ListItem

ListItemCollection

ListItemControlBuilder

Literal

LiteralControlBuilder

MonthChangedEventArgs

PagedDataSource

Panel

PlaceHolder

PlaceHolderControlBuilder

RadioButton

RadioButtonList

RangeValidator

RegularExpressionValidator

Repeater

RepeaterCommandEventArgs

RepeaterItem

RepeaterItemCollection

RepeaterItemEventArgs

RepeatInfo

RequiredFieldValidator

SelectedDatesCollection

ServerValidateEventArgs

Style

Table

Table.RowControlCollection

TableCell

TableCellCollection

TableCellControlBuilder

TableHeaderCell

TableItemStyle

TableRow

TableRow.CellControlCollection

TableRowCollection

TableStyle

TargetConverter

TemplateColumn

TextBox

TextBoxControlBuilder

UnitConverter

ValidatedControlConverter

ValidationSummary

WebColorConverter

WebControl

Xml

Interfaces

IRepeatInfoUser

Structures

FontUnit

Unit

Delegates

AdCreatedEventHandler

CommandEventHandler

DataGridCommandEventHandler

DataGridItemEventHandler

DataGridPageChangedEventHandler

DataGridSortCommandEventHandler

DataListCommandEventHandler

DataListItemEventHandler

DayRenderEventHandler

MonthChangedEventHandler

RepeaterCommandEventHandler

RepeaterItemEventHandler

ServerValidateEventHandler

Enumerations

BorderStyle

ButtonColumnType

CalendarSelectionMode

DayNameFormat

FirstDayOfWeek

FontSize

GridLines

HorizontalAlign

ImageAlign

ListItemType

ListSelectionMode

NextPrevFormat

PagerMode

PagerPosition

RepeatDirection

RepeatLayout

TextAlign

TextBoxMode

TitleFormat

UnitType

ValidationCompareOperator

ValidationDataType

ValidationSummaryDisplayMode

ValidatorDisplay

VerticalAlign

AdCreatedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AdCreatedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public AdCreatedEventArgs(
   System.Collections.IDictionary adProperties
);
```

Public Properties

```
public System.Collections.IDictionary get_AdProperties();
public System.String get_AlternateText();
public void set_AlternateText(
   System.String value
);
public System.String get_ImageUrl();
public void set_ImageUrl(
   System.String value
);
public System.String get_NavigateUrl();
public void set_NavigateUrl(
   System.String value
);
```

AdCreatedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void AdCreatedEventHandler(
   System.Object sender,
   System.Web.UI.WebControls.AdCreatedEventArgs e
);
```

AdRotator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "AdvertisementFile") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "AdCreated") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.AdRotatorDesi
gner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") *
/
/** @attribute System.Web.UI.ToolboxDataAttribute(Data = "<{0}:AdRotator runat="server" Heigh
t="60px" Width="468px"></{0}:AdRotator>") */
public class AdRotator
   extends System.Web.UI.WebControls.WebControl
```

Public Constructors

```
public AdRotator();
```

Public Properties

```
public System.String get_AdvertisementFile();
public void set_AdvertisementFile(
   System.String value
);
public System.Web.UI.WebControls.FontInfo get_Font();
public System.String get_KeywordFilter();
public void set_KeywordFilter(
   System.String value
);
public System.String get_Target();
public void set_Target(
   System.String value
);
```

Public Events

```
public void add_AdCreated(
   System.Web.UI.WebControls.AdCreatedEventHandler value
);
public void remove_AdCreated(
   System.Web.UI.WebControls.AdCreatedEventHandler value
);
```

Protected Methods

```
protected System.Web.UI.ControlCollection CreateControlCollection();
protected void OnAdCreated(
   System.Web.UI.WebControls.AdCreatedEventArgs e
);
protected void OnPreRender(
   System.EventArgs e
);
protected void Render(
   System.Web.UI.HtmlTextWriter writer
```

```
);
```

BaseCompareValidator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class BaseCompareValidator
   extends System.Web.UI.WebControls.BaseValidator
```

Public Properties

```
public System.Web.UI.WebControls.ValidationDataType get_Type();
public void set_Type(
   System.Web.UI.WebControls.ValidationDataType value
);
```

Public Methods

```
public static boolean CanConvert(
   System.String text,
   System.Web.UI.WebControls.ValidationDataType type
);
```

Protected Constructors

```
protected BaseCompareValidator();
```

Protected Properties

```
protected static int get_CutoffYear();
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected static boolean Compare(
   System.String leftText,
   System.String rightText,
   System.Web.UI.WebControls.ValidationCompareOperator op,
   System.Web.UI.WebControls.ValidationDataType type
);
protected static boolean Convert(
   System.String text,
   System.Web.UI.WebControls.ValidationDataType type,
   System.Object& value
);
protected boolean DetermineRenderUplevel();
protected static System.String GetDateElementOrder();
protected static int GetFullYear(
   int shortYear
);
```

BaseDataList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "SelectedIndexChanged") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "DataSource") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.BaseDataListD
esigner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
) */
public abstract class BaseDataList
   extends System.Web.UI.WebControls.WebControl
```

Public Constructors

```
public BaseDataList();
```

Public Properties

```
public int get_CellPadding();
public void set_CellPadding(
   int value
);
public int get_CellSpacing();
public void set_CellSpacing(
   int value
);
public System.Web.UI.ControlCollection get_Controls();
public System.String get_DataKeyField();
public void set_DataKeyField(
   System.String value
);
public System.Web.UI.WebControls.DataKeyCollection get_DataKeys();
public System.String get_DataMember();
public void set_DataMember(
   System.String value
);
public System.Object get_DataSource();
public void set_DataSource(
   System.Object value
);
public System.Web.UI.WebControls.GridLines get_GridLines();
public void set_GridLines(
   System.Web.UI.WebControls.GridLines value
);
public System.Web.UI.WebControls.HorizontalAlign get_HorizontalAlign();
public void set_HorizontalAlign(
   System.Web.UI.WebControls.HorizontalAlign value
);
```

Public Methods

```
public void DataBind();
public static boolean IsBindableType(
   System.Type type
);
```

Public Events

```
public void add_SelectedIndexChanged(
   System.EventHandler value
);
public void remove_SelectedIndexChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Collections.ArrayList get_DataKeysArray();
```

Protected Methods

```
protected void AddParsedSubObject(
   System.Object obj
);
protected void CreateChildControls();
protected abstract void CreateControlHierarchy(
   boolean useDataSource
);
protected void OnDataBinding(
   System.EventArgs e
);
protected void OnSelectedIndexChanged(
   System.EventArgs e
);
protected abstract void PrepareControlHierarchy();
protected void Render(
   System.Web.UI.HtmlTextWriter writer
);
```

BaseValidator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "ErrorMessage") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.BaseValidator
Designer, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
") */
public abstract class BaseValidator
   extends System.Web.UI.WebControls.Label
   implements System.Web.UI.IValidator
```

Public Properties

```
public System.String get_ControlToValidate();
public void set_ControlToValidate(
   System.String value
);
public System.Web.UI.WebControls.ValidatorDisplay get_Display();
public void set_Display(
   System.Web.UI.WebControls.ValidatorDisplay value
);
public boolean get_EnableClientScript();
public void set_EnableClientScript(
   boolean value
);
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public final System.String get_ErrorMessage();
public final void set_ErrorMessage(
   System.String value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public final boolean get_IsValid();
public final void set_IsValid(
   boolean value
);
```

Public Methods

```
public static System.ComponentModel.PropertyDescriptor GetValidationProperty(
   System.Object component
);
public final void Validate();
```

Protected Constructors

```
protected BaseValidator();
```

Protected Properties

```
protected boolean get_PropertiesValid();
protected boolean get_RenderUplevel();
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected void CheckControlValidationProperty(
   System.String name,
   System.String propertyName
);
protected boolean ControlPropertiesValid();
protected boolean DetermineRenderUplevel();
protected abstract boolean EvaluateIsValid();
protected System.String GetControlRenderID(
   System.String name
);
protected System.String GetControlValidationValue(
   System.String name
);
protected void OnInit(
   System.EventArgs e
);
protected void OnPreRender(
   System.EventArgs e
);
protected void OnUnload(
   System.EventArgs e
);
protected void RegisterValidatorCommonScript();
protected void RegisterValidatorDeclaration();
protected void Render(
   System.Web.UI.HtmlTextWriter writer
);
```

BorderStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum BorderStyle
```

Members

Dashed

Dotted

Double

Groove

Inset

None

NotSet

Outset

Ridge

Solid

BoundColumn Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BoundColumn
   extends System.Web.UI.WebControls.DataGridColumn
```

Public Constructors

```
public BoundColumn();
```

Public Properties

```
public System.String get_DataField();
public void set_DataField(
   System.String value
);
public System.String get_DataFormatString();
public void set_DataFormatString(
   System.String value
);
public boolean get_ReadOnly();
public void set_ReadOnly(
   boolean value
);
```

Public Methods

```
public void Initialize();
public void InitializeCell(
   System.Web.UI.WebControls.TableCell cell,
   int columnIndex,
   System.Web.UI.WebControls.ListItemType itemType
);
```

Public Fields

```
public static final System.String thisExpr;
```

Protected Methods

```
protected System.String FormatDataValue(
   System.Object dataValue
);
```

Button Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.ButtonDesigne
r, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.Web.UI.DataBindingHandlerAttribute(HandlerTypeName = "System.Web.UI.Des
ign.TextDataBindingHandler, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToke
n=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Click") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.Web.UI.ToolboxDataAttribute(Data = "<{0}:Button runat=server Text="Butt
on"></{0}:Button>") */
public class Button
   extends System.Web.UI.WebControls.WebControl
   implements System.Web.UI.IPostBackEventHandler
```

Public Constructors

```
public Button();
```

Public Properties

```
public boolean get_CausesValidation();
public void set_CausesValidation(
   boolean value
);
public System.String get_CommandArgument();
public void set_CommandArgument(
   System.String value
);
public System.String get_CommandName();
public void set_CommandName(
   System.String value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Events

```
public void add_Click(
   System.EventHandler value
);
public void remove_Click(
   System.EventHandler value
);
public void add_Command(
   System.Web.UI.WebControls.CommandEventHandler value
);
public void remove_Command(
   System.Web.UI.WebControls.CommandEventHandler value
);
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected void OnClick(
   System.EventArgs e
);
protected void OnCommand(
   System.Web.UI.WebControls.CommandEventArgs e
);
protected void RenderContents(
   System.Web.UI.HtmlTextWriter writer
);
```

ButtonColumn Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ButtonColumn
   extends System.Web.UI.WebControls.DataGridColumn
```

Public Constructors

```
public ButtonColumn();
```

Public Properties

```
public System.Web.UI.WebControls.ButtonColumnType get_ButtonType();
public void set_ButtonType(
   System.Web.UI.WebControls.ButtonColumnType value
);
public System.String get_CommandName();
public void set_CommandName(
   System.String value
);
public System.String get_DataTextField();
public void set_DataTextField(
   System.String value
);
public System.String get_DataTextFormatString();
public void set_DataTextFormatString(
   System.String value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Methods

```
public void Initialize();
public void InitializeCell(
   System.Web.UI.WebControls.TableCell cell,
   int columnIndex,
   System.Web.UI.WebControls.ListItemType itemType
);
```

Protected Methods

```
protected System.String FormatDataTextValue(
   System.Object dataTextValue
);
```

ButtonColumnType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ButtonColumnType
```

Members

LinkButton

PushButton

Calendar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "SelectionChanged") */
/** @attribute System.Web.UI.DataBindingHandlerAttribute(HandlerTypeName = "System.Web.UI.Des
ign.WebControls.CalendarDataBindingHandler, System.Design, Version=1.0.3300.0, Culture=neutra
l, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "SelectedDate") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.CalendarDesig
ner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class Calendar
   extends System.Web.UI.WebControls.WebControl
   implements System.Web.UI.IPostBackEventHandler
```

Public Constructors

```
public Calendar();
```

Public Properties

```
public int get_CellPadding();
public void set_CellPadding(
   int value
);
public int get_CellSpacing();
public void set_CellSpacing(
   int value
);
public System.Web.UI.WebControls.TableItemStyle get_DayHeaderStyle();
public System.Web.UI.WebControls.DayNameFormat get_DayNameFormat();
public void set_DayNameFormat(
   System.Web.UI.WebControls.DayNameFormat value
);
public System.Web.UI.WebControls.TableItemStyle get_DayStyle();
public System.Web.UI.WebControls.FirstDayOfWeek get_FirstDayOfWeek();
public void set_FirstDayOfWeek(
   System.Web.UI.WebControls.FirstDayOfWeek value
);
public System.String get_NextMonthText();
public void set_NextMonthText(
   System.String value
);
public System.Web.UI.WebControls.NextPrevFormat get_NextPrevFormat();
public void set_NextPrevFormat(
   System.Web.UI.WebControls.NextPrevFormat value
);
public System.Web.UI.WebControls.TableItemStyle get_NextPrevStyle();
public System.Web.UI.WebControls.TableItemStyle get_OtherMonthDayStyle();
public System.String get_PrevMonthText();
public void set_PrevMonthText(
   System.String value
);
public System.DateTime get_SelectedDate();
public void set_SelectedDate(
   System.DateTime value
);
public System.Web.UI.WebControls.SelectedDatesCollection get_SelectedDates();
```

```
public System.Web.UI.WebControls.TableItemStyle get_SelectedDayStyle();
public System.Web.UI.WebControls.CalendarSelectionMode get_SelectionMode();
public void set_SelectionMode(
   System.Web.UI.WebControls.CalendarSelectionMode value
);
public System.String get_SelectMonthText();
public void set_SelectMonthText(
   System.String value
);
public System.Web.UI.WebControls.TableItemStyle get_SelectorStyle();
public System.String get_SelectWeekText();
public void set_SelectWeekText(
   System.String value
);
public boolean get_ShowDayHeader();
public void set_ShowDayHeader(
   boolean value
);
public boolean get_ShowGridLines();
public void set_ShowGridLines(
   boolean value
);
public boolean get_ShowNextPrevMonth();
public void set_ShowNextPrevMonth(
   boolean value
);
public boolean get_ShowTitle();
public void set_ShowTitle(
   boolean value
);
public System.Web.UI.WebControls.TitleFormat get_TitleFormat();
public void set_TitleFormat(
   System.Web.UI.WebControls.TitleFormat value
);
public System.Web.UI.WebControls.TableItemStyle get_TitleStyle();
public System.Web.UI.WebControls.TableItemStyle get_TodayDayStyle();
public System.DateTime get_TodaysDate();
public void set_TodaysDate(
   System.DateTime value
);
public System.DateTime get_VisibleDate();
public void set_VisibleDate(
   System.DateTime value
);
public System.Web.UI.WebControls.TableItemStyle get_WeekendDayStyle();
```

Public Events

```
public void add_DayRender(
   System.Web.UI.WebControls.DayRenderEventHandler value
);
public void remove_DayRender(
   System.Web.UI.WebControls.DayRenderEventHandler value
);
public void add_SelectionChanged(
   System.EventHandler value
);
public void remove_SelectionChanged(
   System.EventHandler value
);
public void add_VisibleMonthChanged(
   System.Web.UI.WebControls.MonthChangedEventHandler value
);
public void remove_VisibleMonthChanged(
   System.Web.UI.WebControls.MonthChangedEventHandler value
);
```

Protected Methods

```
protected System.Web.UI.ControlCollection CreateControlCollection();
protected boolean HasWeekSelectors(
   System.Web.UI.WebControls.CalendarSelectionMode selectionMode
);
protected void LoadViewState(
   System.Object savedState
);
protected void OnDayRender(
   System.Web.UI.WebControls.TableCell cell,
   System.Web.UI.WebControls.CalendarDay day
);
protected void OnPreRender(
   System.EventArgs e
);
protected void OnSelectionChanged();
protected void OnVisibleMonthChanged(
   System.DateTime newDate,
   System.DateTime previousDate
);
protected void Render(
   System.Web.UI.HtmlTextWriter writer
);
protected System.Object SaveViewState();
protected void TrackViewState();
```

CalendarDay Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CalendarDay
   extends System.Object
```

Public Constructors

```
public CalendarDay(
   System.DateTime date,
   boolean isWeekend,
   boolean isToday,
   boolean isSelected,
   boolean isOtherMonth,
   System.String dayNumberText
);
```

Public Properties

```
public System.DateTime get_Date();
public System.String get_DayNumberText();
public boolean get_IsOtherMonth();
public boolean get_IsSelectable();
public void set_IsSelectable(
   boolean value
);
public boolean get_IsSelected();
public boolean get_IsToday();
public boolean get_IsWeekend();
```

CalendarSelectionMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CalendarSelectionMode
```

Members

Day

DayWeek

DayWeekMonth

None

CheckBox Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.DataBindingHandlerAttribute(HandlerTypeName = "System.Web.UI.Des
ign.TextDataBindingHandler, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToke
n=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "CheckedChanged") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.CheckBoxDesig
ner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
public class CheckBox
   extends System.Web.UI.WebControls.WebControl
   implements System.Web.UI.IPostBackDataHandler
```

Public Constructors

```
public CheckBox();
```

Public Properties

```
public boolean get_AutoPostBack();
public void set_AutoPostBack(
   boolean value
);
public boolean get_Checked();
public void set_Checked(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.Web.UI.WebControls.TextAlign get_TextAlign();
public void set_TextAlign(
   System.Web.UI.WebControls.TextAlign value
);
```

Public Events

```
public void add_CheckedChanged(
   System.EventHandler value
);
public void remove_CheckedChanged(
   System.EventHandler value
);
```

Protected Methods

```
protected void OnCheckedChanged(
   System.EventArgs e
);
protected void OnPreRender(
   System.EventArgs e
```

```
);
protected void Render(
   System.Web.UI.HtmlTextWriter writer
);
```

CheckBoxList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CheckBoxList
   extends System.Web.UI.WebControls.ListControl
   implements System.Web.UI.WebControls.IRepeatInfoUser, System.Web.UI.INamingContainer, Syst
em.Web.UI.IPostBackDataHandler
```

Public Constructors

```
public CheckBoxList();
```

Public Properties

```
public int get_CellPadding();
public void set_CellPadding(
   int value
);
public int get_CellSpacing();
public void set_CellSpacing(
   int value
);
public int get_RepeatColumns();
public void set_RepeatColumns(
   int value
);
public System.Web.UI.WebControls.RepeatDirection get_RepeatDirection();
public void set_RepeatDirection(
   System.Web.UI.WebControls.RepeatDirection value
);
public System.Web.UI.WebControls.RepeatLayout get_RepeatLayout();
public void set_RepeatLayout(
   System.Web.UI.WebControls.RepeatLayout value
);
public System.Web.UI.WebControls.TextAlign get_TextAlign();
public void set_TextAlign(
   System.Web.UI.WebControls.TextAlign value
);
```

Protected Methods

```
protected System.Web.UI.WebControls.Style CreateControlStyle();
protected System.Web.UI.Control FindControl(
   System.String id,
   int pathOffset
);
protected void OnPreRender(
   System.EventArgs e
);
protected void Render(
   System.Web.UI.HtmlTextWriter writer
);
```

CommandEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CommandEventArgs
   extends System.EventArgs
```

Public Constructors

```
public CommandEventArgs(
   System.Web.UI.WebControls.CommandEventArgs e
);
public CommandEventArgs(
   System.String commandName,
   System.Object argument
);
```

Public Properties

```
public System.Object get_CommandArgument();
public System.String get_CommandName();
```

CommandEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void CommandEventHandler(
   System.Object sender,
   System.Web.UI.WebControls.CommandEventArgs e
);
```

CompareValidator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ToolboxDataAttribute(Data = "<{0}:CompareValidator runat=server
ErrorMessage="CompareValidator"></{0}:CompareValidator>") */
public class CompareValidator
   extends System.Web.UI.WebControls.BaseCompareValidator
```

Public Constructors

```
public CompareValidator();
```

Public Properties

```
public System.String get_ControlToCompare();
public void set_ControlToCompare(
   System.String value
);
public System.Web.UI.WebControls.ValidationCompareOperator get_Operator();
public void set_Operator(
   System.Web.UI.WebControls.ValidationCompareOperator value
);
public System.String get_ValueToCompare();
public void set_ValueToCompare(
   System.String value
);
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected boolean ControlPropertiesValid();
protected boolean EvaluateIsValid();
```

CustomValidator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ToolboxDataAttribute(Data = "<{0}:CustomValidator runat=server E
rrorMessage="CustomValidator"></{0}:CustomValidator>") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ServerValidate") */
public class CustomValidator
   extends System.Web.UI.WebControls.BaseValidator
```

Public Constructors

```
public CustomValidator();
```

Public Properties

```
public System.String get_ClientValidationFunction();
public void set_ClientValidationFunction(
   System.String value
);
```

Public Events

```
public void add_ServerValidate(
   System.Web.UI.WebControls.ServerValidateEventHandler value
);
public void remove_ServerValidate(
   System.Web.UI.WebControls.ServerValidateEventHandler value
);
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected boolean ControlPropertiesValid();
protected boolean EvaluateIsValid();
protected boolean OnServerValidate(
   System.String value
);
```

DataGrid Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.ComponentMo
del.ComponentEditor, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934
e089", EditorTypeName = "System.Web.UI.Design.WebControls.DataGridComponentEditor, System.Des
ign, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.DataGridDesig
ner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class DataGrid
   extends System.Web.UI.WebControls.BaseDataList
   implements System.Web.UI.INamingContainer
```

Public Constructors

```
public DataGrid();
```

Public Properties

```
public boolean get_AllowCustomPaging();
public void set_AllowCustomPaging(
   boolean value
);
public boolean get_AllowPaging();
public void set_AllowPaging(
   boolean value
);
public boolean get_AllowSorting();
public void set_AllowSorting(
   boolean value
);
public System.Web.UI.WebControls.TableItemStyle get_AlternatingItemStyle();
public boolean get_AutoGenerateColumns();
public void set_AutoGenerateColumns(
   boolean value
);
public System.String get_BackImageUrl();
public void set_BackImageUrl(
   System.String value
);
public System.Web.UI.WebControls.DataGridColumnCollection get_Columns();
public int get_CurrentPageIndex();
public void set_CurrentPageIndex(
   int value
);
public int get_EditItemIndex();
public void set_EditItemIndex(
   int value
);
public System.Web.UI.WebControls.TableItemStyle get_EditItemStyle();
public System.Web.UI.WebControls.TableItemStyle get_FooterStyle();
public System.Web.UI.WebControls.TableItemStyle get_HeaderStyle();
public System.Web.UI.WebControls.DataGridItemCollection get_Items();
public System.Web.UI.WebControls.TableItemStyle get_ItemStyle();
public int get_PageCount();
public System.Web.UI.WebControls.DataGridPagerStyle get_PagerStyle();
public int get_PageSize();
```

```
public void set_PageSize(
   int value
);
public int get_SelectedIndex();
public void set_SelectedIndex(
   int value
);
public System.Web.UI.WebControls.DataGridItem get_SelectedItem();
public System.Web.UI.WebControls.TableItemStyle get_SelectedItemStyle();
public boolean get_ShowFooter();
public void set_ShowFooter(
   boolean value
);
public boolean get_ShowHeader();
public void set_ShowHeader(
   boolean value
);
public int get_VirtualItemCount();
public void set_VirtualItemCount(
   int value
);
```

Public Events

```
public void add_CancelCommand(
   System.Web.UI.WebControls.DataGridCommandEventHandler value
);
public void remove_CancelCommand(
   System.Web.UI.WebControls.DataGridCommandEventHandler value
);
public void add_DeleteCommand(
   System.Web.UI.WebControls.DataGridCommandEventHandler value
);
public void remove_DeleteCommand(
   System.Web.UI.WebControls.DataGridCommandEventHandler value
);
public void add_EditCommand(
   System.Web.UI.WebControls.DataGridCommandEventHandler value
);
public void remove_EditCommand(
   System.Web.UI.WebControls.DataGridCommandEventHandler value
);
public void add_ItemCommand(
   System.Web.UI.WebControls.DataGridCommandEventHandler value
);
public void remove_ItemCommand(
   System.Web.UI.WebControls.DataGridCommandEventHandler value
);
public void add_ItemCreated(
   System.Web.UI.WebControls.DataGridItemEventHandler value
);
public void remove_ItemCreated(
   System.Web.UI.WebControls.DataGridItemEventHandler value
);
public void add_ItemDataBound(
   System.Web.UI.WebControls.DataGridItemEventHandler value
);
public void remove_ItemDataBound(
   System.Web.UI.WebControls.DataGridItemEventHandler value
);
public void add_PageIndexChanged(
   System.Web.UI.WebControls.DataGridPageChangedEventHandler value
);
public void remove_PageIndexChanged(
   System.Web.UI.WebControls.DataGridPageChangedEventHandler value
);
```

```
public void add_SortCommand(
   System.Web.UI.WebControls.DataGridSortCommandEventHandler value
);
public void remove_SortCommand(
   System.Web.UI.WebControls.DataGridSortCommandEventHandler value
);
public void add_UpdateCommand(
   System.Web.UI.WebControls.DataGridCommandEventHandler value
);
public void remove_UpdateCommand(
   System.Web.UI.WebControls.DataGridCommandEventHandler value
);
```

Public Fields

```
public static System.String CancelCommandName;
public static System.String DeleteCommandName;
public static System.String EditCommandName;
public static System.String NextPageCommandArgument;
public static System.String PageCommandName;
public static System.String PrevPageCommandArgument;
public static System.String SelectCommandName;
public static System.String SortCommandName;
public static System.String UpdateCommandName;
```

Protected Methods

```
protected System.Collections.ArrayList CreateColumnSet(
   System.Web.UI.WebControls.PagedDataSource dataSource,
   boolean useDataSource
);
protected void CreateControlHierarchy(
   boolean useDataSource
);
protected System.Web.UI.WebControls.Style CreateControlStyle();
protected System.Web.UI.WebControls.DataGridItem CreateItem(
   int itemIndex,
   int dataSourceIndex,
   System.Web.UI.WebControls.ListItemType itemType
);
protected void InitializeItem(
   System.Web.UI.WebControls.DataGridItem item,
   System.Web.UI.WebControls.DataGridColumn[] columns
);
protected void InitializePager(
   System.Web.UI.WebControls.DataGridItem item,
   int columnSpan,
   System.Web.UI.WebControls.PagedDataSource pagedDataSource
);
protected void LoadViewState(
   System.Object savedState
);
protected boolean OnBubbleEvent(
   System.Object source,
   System.EventArgs e
);
protected void OnCancelCommand(
   System.Web.UI.WebControls.DataGridCommandEventArgs e
);
protected void OnDeleteCommand(
   System.Web.UI.WebControls.DataGridCommandEventArgs e
);
protected void OnEditCommand(
   System.Web.UI.WebControls.DataGridCommandEventArgs e
```

```
);
protected void OnItemCommand(
   System.Web.UI.WebControls.DataGridCommandEventArgs e
);
protected void OnItemCreated(
   System.Web.UI.WebControls.DataGridItemEventArgs e
);
protected void OnItemDataBound(
   System.Web.UI.WebControls.DataGridItemEventArgs e
);
protected void OnPageIndexChanged(
   System.Web.UI.WebControls.DataGridPageChangedEventArgs e
);
protected void OnSortCommand(
   System.Web.UI.WebControls.DataGridSortCommandEventArgs e
);
protected void OnUpdateCommand(
   System.Web.UI.WebControls.DataGridCommandEventArgs e
);
protected void PrepareControlHierarchy();
protected System.Object SaveViewState();
protected void TrackViewState();
```

DataGridColumn Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Compo
nentModel.ExpandableObjectConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyTo
ken=b77a5c561934e089") */
public abstract class DataGridColumn
   extends System.Object
   implements System.Web.UI.IStateManager
```

Public Constructors

```
public DataGridColumn();
```

Public Properties

```
public System.Web.UI.WebControls.TableItemStyle get_FooterStyle();
public System.String get_FooterText();
public void set_FooterText(
   System.String value
);
public System.String get_HeaderImageUrl();
public void set_HeaderImageUrl(
   System.String value
);
public System.Web.UI.WebControls.TableItemStyle get_HeaderStyle();
public System.String get_HeaderText();
public void set_HeaderText(
   System.String value
);
public System.Web.UI.WebControls.TableItemStyle get_ItemStyle();
public System.String get_SortExpression();
public void set_SortExpression(
   System.String value
);
public boolean get_Visible();
public void set_Visible(
   boolean value
);
```

Public Methods

```
public void Initialize();
public void InitializeCell(
   System.Web.UI.WebControls.TableCell cell,
   int columnIndex,
   System.Web.UI.WebControls.ListItemType itemType
);
public System.String ToString();
```

Protected Properties

```
protected boolean get_DesignMode();
protected boolean get_IsTrackingViewState();
```

```
protected System.Web.UI.WebControls.DataGrid get_Owner();
protected System.Web.UI.StateBag get_ViewState();
```

Protected Methods

```
protected void LoadViewState(
   System.Object savedState
);
protected void OnColumnChanged();
protected System.Object SaveViewState();
protected void TrackViewState();
```

DataGridColumnCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DataGridColumnCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable, System.Web.UI.I
StateManager
```

Public Constructors

```
public DataGridColumnCollection(
   System.Web.UI.WebControls.DataGrid owner,
   System.Collections.ArrayList columns
);
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Web.UI.WebControls.DataGridColumn get_Item(
   int index
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public void Add(
   System.Web.UI.WebControls.DataGridColumn column
);
public void AddAt(
   int index,
   System.Web.UI.WebControls.DataGridColumn column
);
public void Clear();
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Web.UI.WebControls.DataGridColumn column
);
public void Remove(
   System.Web.UI.WebControls.DataGridColumn column
);
public void RemoveAt(
   int index
);
```

DataGridCommandEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataGridCommandEventArgs
   extends System.Web.UI.WebControls.CommandEventArgs
```

Public Constructors

```
public DataGridCommandEventArgs(
   System.Web.UI.WebControls.DataGridItem item,
   System.Object commandSource,
   System.Web.UI.WebControls.CommandEventArgs originalArgs
);
```

Public Properties

```
public System.Object get_CommandSource();
public System.Web.UI.WebControls.DataGridItem get_Item();
```

DataGridCommandEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DataGridCommandEventHandler(
   System.Object source,
   System.Web.UI.WebControls.DataGridCommandEventArgs e
);
```

DataGridItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataGridItem
   extends System.Web.UI.WebControls.TableRow
   implements System.Web.UI.INamingContainer
```

Public Constructors

```
public DataGridItem(
   int itemIndex,
   int dataSetIndex,
   System.Web.UI.WebControls.ListItemType itemType
);
```

Public Properties

```
public System.Object get_DataItem();
public void set_DataItem(
   System.Object value
);
public int get_DataSetIndex();
public int get_ItemIndex();
public System.Web.UI.WebControls.ListItemType get_ItemType();
```

Protected Methods

```
protected boolean OnBubbleEvent(
   System.Object source,
   System.EventArgs e
);
protected void SetItemType(
   System.Web.UI.WebControls.ListItemType itemType
);
```

DataGridItemCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DataGridItemCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public DataGridItemCollection(
   System.Collections.ArrayList items
);
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Web.UI.WebControls.DataGridItem get_Item(
   int index
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
```

DataGridItemEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataGridItemEventArgs
   extends System.EventArgs
```

Public Constructors

```
public DataGridItemEventArgs(
   System.Web.UI.WebControls.DataGridItem item
);
```

Public Properties

```
public System.Web.UI.WebControls.DataGridItem get_Item();
```

DataGridItemEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DataGridItemEventHandler(
   System.Object sender,
   System.Web.UI.WebControls.DataGridItemEventArgs e
);
```

DataGridPageChangedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataGridPageChangedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public DataGridPageChangedEventArgs(
   System.Object commandSource,
   int newPageIndex
);
```

Public Properties

```
public System.Object get_CommandSource();
public int get_NewPageIndex();
```

DataGridPageChangedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DataGridPageChangedEventHandler(
   System.Object source,
   System.Web.UI.WebControls.DataGridPageChangedEventArgs e
);
```

DataGridPagerStyle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataGridPagerStyle
   extends System.Web.UI.WebControls.TableItemStyle
```

Public Properties

```
public System.Web.UI.WebControls.PagerMode get_Mode();
public void set_Mode(
   System.Web.UI.WebControls.PagerMode value
);
public System.String get_NextPageText();
public void set_NextPageText(
   System.String value
);
public int get_PageButtonCount();
public void set_PageButtonCount(
   int value
);
public System.Web.UI.WebControls.PagerPosition get_Position();
public void set_Position(
   System.Web.UI.WebControls.PagerPosition value
);
public System.String get_PrevPageText();
public void set_PrevPageText(
   System.String value
);
public boolean get_Visible();
public void set_Visible(
   boolean value
);
```

Public Methods

```
public void CopyFrom(
   System.Web.UI.WebControls.Style s
);
public void MergeWith(
   System.Web.UI.WebControls.Style s
);
public void Reset();
```

DataGridSortCommandEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataGridSortCommandEventArgs
   extends System.EventArgs
```

Public Constructors

```
public DataGridSortCommandEventArgs(
   System.Object commandSource,
   System.Web.UI.WebControls.DataGridCommandEventArgs dce
);
```

Public Properties

```
public System.Object get_CommandSource();
public System.String get_SortExpression();
```

DataGridSortCommandEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DataGridSortCommandEventHandler(
   System.Object source,
   System.Web.UI.WebControls.DataGridSortCommandEventArgs e
);
```

DataKeyCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DataKeyCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public DataKeyCollection(
   System.Collections.ArrayList keys
);
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Object get_Item(
   int index
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
```

DataList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.ComponentMo
del.ComponentEditor, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c561934
e089", EditorTypeName = "System.Web.UI.Design.WebControls.DataListComponentEditor, System.Des
ign, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.DataListDesig
ner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class DataList
   extends System.Web.UI.WebControls.BaseDataList
   implements System.Web.UI.INamingContainer, System.Web.UI.WebControls.IRepeatInfoUser
```

Public Constructors

```
public DataList();
```

Public Properties

```
public System.Web.UI.WebControls.TableItemStyle get_AlternatingItemStyle();
public System.Web.UI.ITemplate get_AlternatingItemTemplate();
public void set_AlternatingItemTemplate(
   System.Web.UI.ITemplate value
);
public int get_EditItemIndex();
public void set_EditItemIndex(
   int value
);
public System.Web.UI.WebControls.TableItemStyle get_EditItemStyle();
public System.Web.UI.ITemplate get_EditItemTemplate();
public void set_EditItemTemplate(
   System.Web.UI.ITemplate value
);
public boolean get_ExtractTemplateRows();
public void set_ExtractTemplateRows(
   boolean value
);
public System.Web.UI.WebControls.TableItemStyle get_FooterStyle();
public System.Web.UI.ITemplate get_FooterTemplate();
public void set_FooterTemplate(
   System.Web.UI.ITemplate value
);
public System.Web.UI.WebControls.GridLines get_GridLines();
public void set_GridLines(
   System.Web.UI.WebControls.GridLines value
);
public System.Web.UI.WebControls.TableItemStyle get_HeaderStyle();
public System.Web.UI.ITemplate get_HeaderTemplate();
public void set_HeaderTemplate(
   System.Web.UI.ITemplate value
);
public System.Web.UI.WebControls.DataListItemCollection get_Items();
public System.Web.UI.WebControls.TableItemStyle get_ItemStyle();
public System.Web.UI.ITemplate get_ItemTemplate();
public void set_ItemTemplate(
   System.Web.UI.ITemplate value
);
```

```
public int get_RepeatColumns();
public void set_RepeatColumns(
   int value
);
public System.Web.UI.WebControls.RepeatDirection get_RepeatDirection();
public void set_RepeatDirection(
   System.Web.UI.WebControls.RepeatDirection value
);
public System.Web.UI.WebControls.RepeatLayout get_RepeatLayout();
public void set_RepeatLayout(
   System.Web.UI.WebControls.RepeatLayout value
);
public int get_SelectedIndex();
public void set_SelectedIndex(
   int value
);
public System.Web.UI.WebControls.DataListItem get_SelectedItem();
public System.Web.UI.WebControls.TableItemStyle get_SelectedItemStyle();
public System.Web.UI.ITemplate get_SelectedItemTemplate();
public void set_SelectedItemTemplate(
   System.Web.UI.ITemplate value
);
public System.Web.UI.WebControls.TableItemStyle get_SeparatorStyle();
public System.Web.UI.ITemplate get_SeparatorTemplate();
public void set_SeparatorTemplate(
   System.Web.UI.ITemplate value
);
public boolean get_ShowFooter();
public void set_ShowFooter(
   boolean value
);
public boolean get_ShowHeader();
public void set_ShowHeader(
   boolean value
);
```

Public Events

```
public void add_CancelCommand(
   System.Web.UI.WebControls.DataListCommandEventHandler value
);
public void remove_CancelCommand(
   System.Web.UI.WebControls.DataListCommandEventHandler value
);
public void add_DeleteCommand(
   System.Web.UI.WebControls.DataListCommandEventHandler value
);
public void remove_DeleteCommand(
   System.Web.UI.WebControls.DataListCommandEventHandler value
);
public void add_EditCommand(
   System.Web.UI.WebControls.DataListCommandEventHandler value
);
public void remove_EditCommand(
   System.Web.UI.WebControls.DataListCommandEventHandler value
);
public void add_ItemCommand(
   System.Web.UI.WebControls.DataListCommandEventHandler value
);
public void remove_ItemCommand(
   System.Web.UI.WebControls.DataListCommandEventHandler value
);
public void add_ItemCreated(
   System.Web.UI.WebControls.DataListItemEventHandler value
);
public void remove_ItemCreated(
```

```
   System.Web.UI.WebControls.DataListItemEventHandler value
);
public void add_ItemDataBound(
   System.Web.UI.WebControls.DataListItemEventHandler value
);
public void remove_ItemDataBound(
   System.Web.UI.WebControls.DataListItemEventHandler value
);
public void add_UpdateCommand(
   System.Web.UI.WebControls.DataListCommandEventHandler value
);
public void remove_UpdateCommand(
   System.Web.UI.WebControls.DataListCommandEventHandler value
);
```

Public Fields

```
public static System.String CancelCommandName;
public static System.String DeleteCommandName;
public static System.String EditCommandName;
public static System.String SelectCommandName;
public static System.String UpdateCommandName;
```

Protected Methods

```
protected void CreateControlHierarchy(
   boolean useDataSource
);
protected System.Web.UI.WebControls.Style CreateControlStyle();
protected System.Web.UI.WebControls.DataListItem CreateItem(
   int itemIndex,
   System.Web.UI.WebControls.ListItemType itemType
);
protected void InitializeItem(
   System.Web.UI.WebControls.DataListItem item
);
protected void LoadViewState(
   System.Object savedState
);
protected boolean OnBubbleEvent(
   System.Object source,
   System.EventArgs e
);
protected void OnCancelCommand(
   System.Web.UI.WebControls.DataListCommandEventArgs e
);
protected void OnDeleteCommand(
   System.Web.UI.WebControls.DataListCommandEventArgs e
);
protected void OnEditCommand(
   System.Web.UI.WebControls.DataListCommandEventArgs e
);
protected void OnItemCommand(
   System.Web.UI.WebControls.DataListCommandEventArgs e
);
protected void OnItemCreated(
   System.Web.UI.WebControls.DataListItemEventArgs e
);
protected void OnItemDataBound(
   System.Web.UI.WebControls.DataListItemEventArgs e
);
protected void OnUpdateCommand(
   System.Web.UI.WebControls.DataListCommandEventArgs e
);
```

```
protected void PrepareControlHierarchy();
protected void RenderContents(
   System.Web.UI.HtmlTextWriter writer
);
protected System.Object SaveViewState();
protected void TrackViewState();
```

DataListCommandEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataListCommandEventArgs
   extends System.Web.UI.WebControls.CommandEventArgs
```

Public Constructors

```
public DataListCommandEventArgs(
   System.Web.UI.WebControls.DataListItem item,
   System.Object commandSource,
   System.Web.UI.WebControls.CommandEventArgs originalArgs
);
```

Public Properties

```
public System.Object get_CommandSource();
public System.Web.UI.WebControls.DataListItem get_Item();
```

DataListCommandEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DataListCommandEventHandler(
   System.Object source,
   System.Web.UI.WebControls.DataListCommandEventArgs e
);
```

DataListItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class DataListItem
   extends System.Web.UI.WebControls.WebControl
   implements System.Web.UI.INamingContainer
```

Public Constructors

```
public DataListItem(
   int itemIndex,
   System.Web.UI.WebControls.ListItemType itemType
);
```

Public Properties

```
public System.Object get_DataItem();
public void set_DataItem(
   System.Object value
);
public int get_ItemIndex();
public System.Web.UI.WebControls.ListItemType get_ItemType();
```

Public Methods

```
public void RenderItem(
   System.Web.UI.HtmlTextWriter writer,
   boolean extractRows,
   boolean tableLayout
);
```

Protected Methods

```
protected System.Web.UI.WebControls.Style CreateControlStyle();
protected boolean OnBubbleEvent(
   System.Object source,
   System.EventArgs e
);
protected void SetItemType(
   System.Web.UI.WebControls.ListItemType itemType
);
```

DataListItemCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DataListItemCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public DataListItemCollection(
   System.Collections.ArrayList items
);
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Web.UI.WebControls.DataListItem get_Item(
   int index
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
```

DataListItemEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataListItemEventArgs
   extends System.EventArgs
```

Public Constructors

```
public DataListItemEventArgs(
   System.Web.UI.WebControls.DataListItem item
);
```

Public Properties

```
public System.Web.UI.WebControls.DataListItem get_Item();
```

DataListItemEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DataListItemEventHandler(
   System.Object sender,
   System.Web.UI.WebControls.DataListItemEventArgs e
);
```

DayNameFormat Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum DayNameFormat
```

Members

FirstLetter

FirstTwoLetters

Full

Short

DayRenderEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DayRenderEventArgs
   extends System.Object
```

Public Constructors

```
public DayRenderEventArgs(
   System.Web.UI.WebControls.TableCell cell,
   System.Web.UI.WebControls.CalendarDay day
);
```

Public Properties

```
public System.Web.UI.WebControls.TableCell get_Cell();
public System.Web.UI.WebControls.CalendarDay get_Day();
```

DayRenderEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DayRenderEventHandler(
   System.Object sender,
   System.Web.UI.WebControls.DayRenderEventArgs e
);
```

DropDownList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ValidationPropertyAttribute(Name = "SelectedItem") */
public class DropDownList
   extends System.Web.UI.WebControls.ListControl
   implements System.Web.UI.IPostBackDataHandler
```

Public Constructors

```
public DropDownList();
```

Public Properties

```
public System.Drawing.Color get_BorderColor();
public void set_BorderColor(
   System.Drawing.Color value
);
public System.Web.UI.WebControls.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Web.UI.WebControls.BorderStyle value
);
public System.Web.UI.WebControls.Unit get_BorderWidth();
public void set_BorderWidth(
   System.Web.UI.WebControls.Unit value
);
public int get_SelectedIndex();
public void set_SelectedIndex(
   int value
);
public System.String get_ToolTip();
public void set_ToolTip(
   System.String value
);
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected System.Web.UI.ControlCollection CreateControlCollection();
protected void RenderContents(
   System.Web.UI.HtmlTextWriter writer
);
```

EditCommandColumn Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EditCommandColumn
   extends System.Web.UI.WebControls.DataGridColumn
```

Public Constructors

```
public EditCommandColumn();
```

Public Properties

```
public System.Web.UI.WebControls.ButtonColumnType get_ButtonType();
public void set_ButtonType(
   System.Web.UI.WebControls.ButtonColumnType value
);
public System.String get_CancelText();
public void set_CancelText(
   System.String value
);
public System.String get_EditText();
public void set_EditText(
   System.String value
);
public System.String get_UpdateText();
public void set_UpdateText(
   System.String value
);
```

Public Methods

```
public void InitializeCell(
   System.Web.UI.WebControls.TableCell cell,
   int columnIndex,
   System.Web.UI.WebControls.ListItemType itemType
);
```

FirstDayOfWeek Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum FirstDayOfWeek
```

Members

Default

Friday

Monday

Saturday

Sunday

Thursday

Tuesday

Wednesday

FontInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Compo
nentModel.ExpandableObjectConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyTo
ken=b77a5c561934e089") */
public class FontInfo
   extends System.Object
```

Public Properties

```
public boolean get_Bold();
public void set_Bold(
   boolean value
);
public boolean get_Italic();
public void set_Italic(
   boolean value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String[] get_Names();
public void set_Names(
   System.String[] value
);
public boolean get_Overline();
public void set_Overline(
   boolean value
);
public System.Web.UI.WebControls.FontUnit get_Size();
public void set_Size(
   System.Web.UI.WebControls.FontUnit value
);
public boolean get_Strikeout();
public void set_Strikeout(
   boolean value
);
public boolean get_Underline();
public void set_Underline(
   boolean value
);
```

Public Methods

```
public void CopyFrom(
   System.Web.UI.WebControls.FontInfo f
);
public void MergeWith(
   System.Web.UI.WebControls.FontInfo f
);
public boolean ShouldSerializeNames();
public System.String ToString();
```

FontNamesConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FontNamesConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public FontNamesConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

FontSize Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum FontSize
```

Members

AsUnit

Large

Larger

Medium

NotSet

Small

Smaller

XLarge

XSmall

XXLarge

XXSmall

FontUnit Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Web.U
I.WebControls.FontUnitConverter, System.Web, Version=1.0.3300.0, Culture=neutral, PublicKeyTo
ken=b03f5f7f11d50a3a") */
public class FontUnit
   extends System.ValueType
```

Public Constructors

```
public FontUnit(
   int value
);
public FontUnit(
   System.String value
);
public FontUnit(
   System.Web.UI.WebControls.FontSize type
);
public FontUnit(
   System.Web.UI.WebControls.Unit value
);
public FontUnit(
   System.String value,
   System.Globalization.CultureInfo culture
);
```

Public Properties

```
public boolean get_IsEmpty();
public System.Web.UI.WebControls.FontSize get_Type();
public System.Web.UI.WebControls.Unit get_Unit();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public static System.Web.UI.WebControls.FontUnit Parse(
   System.String s
);
public static System.Web.UI.WebControls.FontUnit Parse(
   System.String s,
   System.Globalization.CultureInfo culture
);
public static System.Web.UI.WebControls.FontUnit Point(
   int n
);
public System.String ToString();
public System.String ToString(
   System.Globalization.CultureInfo culture
);
```

Public Operators

```
public static boolean op_Equality(
   System.Web.UI.WebControls.FontUnit left,
   System.Web.UI.WebControls.FontUnit right
);
public static System.Web.UI.WebControls.FontUnit op_Implicit(
   int n
);
public static boolean op_Inequality(
   System.Web.UI.WebControls.FontUnit left,
   System.Web.UI.WebControls.FontUnit right
);
```

Public Fields

```
public static final System.Web.UI.WebControls.FontUnit Empty;
public static final System.Web.UI.WebControls.FontUnit Large;
public static final System.Web.UI.WebControls.FontUnit Larger;
public static final System.Web.UI.WebControls.FontUnit Medium;
public static final System.Web.UI.WebControls.FontUnit Small;
public static final System.Web.UI.WebControls.FontUnit Smaller;
public static final System.Web.UI.WebControls.FontUnit XLarge;
public static final System.Web.UI.WebControls.FontUnit XSmall;
public static final System.Web.UI.WebControls.FontUnit XXLarge;
public static final System.Web.UI.WebControls.FontUnit XXSmall;
```

FontUnitConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FontUnitConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public FontUnitConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

GridLines Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum GridLines
```

Members

Both

Horizontal

None

Vertical

HorizontalAlign Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Web.U
I.WebControls.HorizontalAlignConverter, System.Web, Version=1.0.3300.0, Culture=neutral, Publ
icKeyToken=b03f5f7f11d50a3a") */
public enum HorizontalAlign
```

Members

Center

Justify

Left

NotSet

Right

HyperLink Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ParseChildrenAttribute(ChildrenAsProperties = false, DefaultProp
erty = "") */
/** @attribute System.Web.UI.DataBindingHandlerAttribute(HandlerTypeName = "System.Web.UI.Des
ign.HyperLinkDataBindingHandler, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKe
yToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.HyperLinkDesi
gner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") *
/
/** @attribute System.Web.UI.ToolboxDataAttribute(Data = "<{0}:HyperLink runat=server>HyperLi
nk</{0}:HyperLink>") */
/** @attribute System.Web.UI.ControlBuilderAttribute(BuilderType = System.Web.UI.WebControls.
HyperLinkControlBuilder) */
public class HyperLink
   extends System.Web.UI.WebControls.WebControl
```

Public Constructors

```
public HyperLink();
```

Public Properties

```
public System.String get_ImageUrl();
public void set_ImageUrl(
   System.String value
);
public System.String get_NavigateUrl();
public void set_NavigateUrl(
   System.String value
);
public System.String get_Target();
public void set_Target(
   System.String value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected void AddParsedSubObject(
   System.Object obj
);
protected void LoadViewState(
   System.Object savedState
);
protected void RenderContents(
   System.Web.UI.HtmlTextWriter writer
```

```
);
```

HyperLinkColumn Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HyperLinkColumn
   extends System.Web.UI.WebControls.DataGridColumn
```

Public Constructors

```
public HyperLinkColumn();
```

Public Properties

```
public System.String get_DataNavigateUrlField();
public void set_DataNavigateUrlField(
   System.String value
);
public System.String get_DataNavigateUrlFormatString();
public void set_DataNavigateUrlFormatString(
   System.String value
);
public System.String get_DataTextField();
public void set_DataTextField(
   System.String value
);
public System.String get_DataTextFormatString();
public void set_DataTextFormatString(
   System.String value
);
public System.String get_NavigateUrl();
public void set_NavigateUrl(
   System.String value
);
public System.String get_Target();
public void set_Target(
   System.String value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Methods

```
public void Initialize();
public void InitializeCell(
   System.Web.UI.WebControls.TableCell cell,
   int columnIndex,
   System.Web.UI.WebControls.ListItemType itemType
);
```

Protected Methods

```
protected System.String FormatDataNavigateUrlValue(
   System.Object dataUrlValue
```

```
);
protected System.String FormatDataTextValue(
   System.Object dataTextValue
);
```

HyperLinkControlBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HyperLinkControlBuilder
   extends System.Web.UI.ControlBuilder
```

Public Constructors

```
public HyperLinkControlBuilder();
```

Public Methods

```
public boolean AllowWhitespaceLiterals();
```

Image Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "ImageUrl") */
public class Image
   extends System.Web.UI.WebControls.WebControl
```

Public Constructors

```
public Image();
```

Public Properties

```
public System.String get_AlternateText();
public void set_AlternateText(
   System.String value
);
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public System.Web.UI.WebControls.FontInfo get_Font();
public System.Web.UI.WebControls.ImageAlign get_ImageAlign();
public void set_ImageAlign(
   System.Web.UI.WebControls.ImageAlign value
);
public System.String get_ImageUrl();
public void set_ImageUrl(
   System.String value
);
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected void RenderContents(
   System.Web.UI.HtmlTextWriter writer
);
```

ImageAlign Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ImageAlign
```

Members

AbsBottom

AbsMiddle

Baseline

Bottom

Left

Middle

NotSet

Right

TextTop

Top

ImageButton Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Click") */
public class ImageButton
   extends System.Web.UI.WebControls.Image
   implements System.Web.UI.IPostBackDataHandler, System.Web.UI.IPostBackEventHandler
```

Public Constructors

```
public ImageButton();
```

Public Properties

```
public boolean get_CausesValidation();
public void set_CausesValidation(
   boolean value
);
public System.String get_CommandArgument();
public void set_CommandArgument(
   System.String value
);
public System.String get_CommandName();
public void set_CommandName(
   System.String value
);
```

Public Events

```
public void add_Click(
   System.Web.UI.ImageClickEventHandler value
);
public void remove_Click(
   System.Web.UI.ImageClickEventHandler value
);
public void add_Command(
   System.Web.UI.WebControls.CommandEventHandler value
);
public void remove_Command(
   System.Web.UI.WebControls.CommandEventHandler value
);
```

Protected Properties

```
protected System.Web.UI.HtmlTextWriterTag get_TagKey();
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected void OnClick(
```

```
    System.Web.UI.ImageClickEventArgs e
);
protected void OnCommand(
    System.Web.UI.WebControls.CommandEventArgs e
);
protected void OnPreRender(
    System.EventArgs e
);
```

IRepeatInfoUser Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IRepeatInfoUser
```

Public Properties

```
boolean get_HasFooter();
boolean get_HasHeader();
boolean get_HasSeparators();
int get_RepeatedItemCount();
```

Public Methods

```
System.Web.UI.WebControls.Style GetItemStyle(
   System.Web.UI.WebControls.ListItemType itemType,
   int repeatIndex
);
void RenderItem(
   System.Web.UI.WebControls.ListItemType itemType,
   int repeatIndex,
   System.Web.UI.WebControls.RepeatInfo repeatInfo,
   System.Web.UI.HtmlTextWriter writer
);
```

Label Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ParseChildrenAttribute(ChildrenAsProperties = false, DefaultProp
erty = "") */
/** @attribute System.Web.UI.ControlBuilderAttribute(BuilderType = System.Web.UI.WebControls.
LabelControlBuilder) */
/** @attribute System.Web.UI.DataBindingHandlerAttribute(HandlerTypeName = "System.Web.UI.Des
ign.TextDataBindingHandler, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToke
n=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.LabelDesigner
, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.Web.UI.ToolboxDataAttribute(Data = "<{0}:Label runat=server>Label</{0}:
Label>") */
public class Label
   extends System.Web.UI.WebControls.WebControl
```

Public Constructors

```
public Label();
```

Public Properties

```
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Protected Methods

```
protected void AddParsedSubObject(
   System.Object obj
);
protected void LoadViewState(
   System.Object savedState
);
protected void RenderContents(
   System.Web.UI.HtmlTextWriter writer
);
```

LabelControlBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LabelControlBuilder
   extends System.Web.UI.ControlBuilder
```

Public Constructors

```
public LabelControlBuilder();
```

Public Methods

```
public boolean AllowWhitespaceLiterals();
```

LinkButton Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.DataBindingHandlerAttribute(HandlerTypeName = "System.Web.UI.Des
ign.TextDataBindingHandler, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToke
n=b03f5f7f11d50a3a") */
/** @attribute System.Web.UI.ControlBuilderAttribute(BuilderType = System.Web.UI.WebControls.
LinkButtonControlBuilder) */
/** @attribute System.Web.UI.ToolboxDataAttribute(Data = "<{0}:LinkButton runat=server>LinkBu
tton</{0}:LinkButton>") */
/** @attribute System.Web.UI.ParseChildrenAttribute(ChildrenAsProperties = false, DefaultProp
erty = "") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Click") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.LinkButtonDes
igner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")
*/
public class LinkButton
   extends System.Web.UI.WebControls.WebControl
   implements System.Web.UI.IPostBackEventHandler
```

Public Constructors

```
public LinkButton();
```

Public Properties

```
public boolean get_CausesValidation();
public void set_CausesValidation(
   boolean value
);
public System.String get_CommandArgument();
public void set_CommandArgument(
   System.String value
);
public System.String get_CommandName();
public void set_CommandName(
   System.String value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Events

```
public void add_Click(
   System.EventHandler value
);
public void remove_Click(
   System.EventHandler value
);
public void add_Command(
   System.Web.UI.WebControls.CommandEventHandler value
);
```

```
public void remove_Command(
   System.Web.UI.WebControls.CommandEventHandler value
);
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected void AddParsedSubObject(
   System.Object obj
);
protected void LoadViewState(
   System.Object savedState
);
protected void OnClick(
   System.EventArgs e
);
protected void OnCommand(
   System.Web.UI.WebControls.CommandEventArgs e
);
protected void OnPreRender(
   System.EventArgs e
);
protected void RenderContents(
   System.Web.UI.HtmlTextWriter writer
);
```

LinkButtonControlBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LinkButtonControlBuilder
   extends System.Web.UI.ControlBuilder
```

Public Constructors

```
public LinkButtonControlBuilder();
```

Public Methods

```
public boolean AllowWhitespaceLiterals();
```

ListBox Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ValidationPropertyAttribute(Name = "SelectedItem") */
public class ListBox
   extends System.Web.UI.WebControls.ListControl
   implements System.Web.UI.IPostBackDataHandler
```

Public Constructors

```
public ListBox();
```

Public Properties

```
public System.Drawing.Color get_BorderColor();
public void set_BorderColor(
   System.Drawing.Color value
);
public System.Web.UI.WebControls.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Web.UI.WebControls.BorderStyle value
);
public System.Web.UI.WebControls.Unit get_BorderWidth();
public void set_BorderWidth(
   System.Web.UI.WebControls.Unit value
);
public int get_Rows();
public void set_Rows(
   int value
);
public System.Web.UI.WebControls.ListSelectionMode get_SelectionMode();
public void set_SelectionMode(
   System.Web.UI.WebControls.ListSelectionMode value
);
public System.String get_ToolTip();
public void set_ToolTip(
   System.String value
);
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected void OnPreRender(
   System.EventArgs e
);
protected void RenderContents(
   System.Web.UI.HtmlTextWriter writer
);
```

ListControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.ListControlDe
signer, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")
*/
/** @attribute System.Web.UI.DataBindingHandlerAttribute(HandlerTypeName = "System.Web.UI.Des
ign.WebControls.ListControlDataBindingHandler, System.Design, Version=1.0.3300.0, Culture=neu
tral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "SelectedIndexChanged") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "DataSource") */
/** @attribute System.Web.UI.ParseChildrenAttribute(ChildrenAsProperties = true, DefaultProp
erty = "Items") */
public abstract class ListControl
   extends System.Web.UI.WebControls.WebControl
```

Public Constructors

```
public ListControl();
```

Public Properties

```
public boolean get_AutoPostBack();
public void set_AutoPostBack(
   boolean value
);
public System.String get_DataMember();
public void set_DataMember(
   System.String value
);
public System.Object get_DataSource();
public void set_DataSource(
   System.Object value
);
public System.String get_DataTextField();
public void set_DataTextField(
   System.String value
);
public System.String get_DataTextFormatString();
public void set_DataTextFormatString(
   System.String value
);
public System.String get_DataValueField();
public void set_DataValueField(
   System.String value
);
public System.Web.UI.WebControls.ListItemCollection get_Items();
public int get_SelectedIndex();
public void set_SelectedIndex(
   int value
);
public System.Web.UI.WebControls.ListItem get_SelectedItem();
```

Public Methods

```
public void ClearSelection();
```

Public Events

```
public void add_SelectedIndexChanged(
   System.EventHandler value
);
public void remove_SelectedIndexChanged(
   System.EventHandler value
);
```

Protected Methods

```
protected void LoadViewState(
   System.Object savedState
);
protected void OnDataBinding(
   System.EventArgs e
);
protected void OnPreRender(
   System.EventArgs e
);
protected void OnSelectedIndexChanged(
   System.EventArgs e
);
protected System.Object SaveViewState();
protected void TrackViewState();
```

ListItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ControlBuilderAttribute(BuilderType = System.Web.UI.WebControls.
ListItemControlBuilder) */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Compo
nentModel.ExpandableObjectConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyTo
ken=b77a5c561934e089") */
public class ListItem
   extends System.Object
   implements System.Web.UI.IStateManager, System.Web.UI.IParserAccessor, System.Web.UI.IAttr
ibuteAccessor
```

Public Constructors

```
public ListItem();
public ListItem(
   System.String text
);
public ListItem(
   System.String text,
   System.String value
);
```

Public Properties

```
public System.Web.UI.AttributeCollection get_Attributes();
public boolean get_Selected();
public void set_Selected(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Methods

```
public boolean Equals(
   System.Object o
);
public static System.Web.UI.WebControls.ListItem FromString(
   System.String s
);
public int GetHashCode();
public System.String ToString();
```

ListItemCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Web.UI.Design.WebControls.ListItemsCollectionEditor, Sys
tem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ListItemCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable, System.Web.UI.IStateManager
```

Public Constructors

```
public ListItemCollection();
```

Public Properties

```
public int get_Capacity();
public void set_Capacity(
   int value
);
public final int get_Count();
public final boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Web.UI.WebControls.ListItem get_Item(
   int index
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public void Add(
   System.String item
);
public void Add(
   System.Web.UI.WebControls.ListItem item
);
public void AddRange(
   System.Web.UI.WebControls.ListItem[] items
);
public final void Clear();
public boolean Contains(
   System.Web.UI.WebControls.ListItem item
);
public final void CopyTo(
   System.Array array,
   int index
);
public System.Web.UI.WebControls.ListItem FindByText(
   System.String text
);
public System.Web.UI.WebControls.ListItem FindByValue(
   System.String value
);
```

```
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Web.UI.WebControls.ListItem item
);
public void Insert(
   int index,
   System.String item
);
public void Insert(
   int index,
   System.Web.UI.WebControls.ListItem item
);
public void Remove(
   System.String item
);
public void Remove(
   System.Web.UI.WebControls.ListItem item
);
public final void RemoveAt(
   int index
);
```

ListItemControlBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ListItemControlBuilder
   extends System.Web.UI.ControlBuilder
```

Public Constructors

```
public ListItemControlBuilder();
```

Public Methods

```
public boolean AllowWhitespaceLiterals();
public boolean HtmlDecodeLiterals();
```

ListItemType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ListItemType
```

Members

AlternatingItem

EditItem

Footer

Header

Item

Pager

SelectedItem

Separator

ListSelectionMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ListSelectionMode
```

Members

Multiple

Single

Literal Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.DataBindingHandlerAttribute(HandlerTypeName = "System.Web.UI.Des
ign.TextDataBindingHandler, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToke
n=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.Web.UI.ControlBuilderAttribute(BuilderType = System.Web.UI.WebControls.
LiteralControlBuilder) */
public class Literal
   extends System.Web.UI.Control
```

Public Constructors

```
public Literal();
```

Public Properties

```
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Protected Methods

```
protected void AddParsedSubObject(
   System.Object obj
);
protected System.Web.UI.ControlCollection CreateControlCollection();
protected void Render(
   System.Web.UI.HtmlTextWriter output
);
```

LiteralControlBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LiteralControlBuilder
   extends System.Web.UI.ControlBuilder
```

Public Constructors

```
public LiteralControlBuilder();
```

Public Methods

```
public boolean AllowWhitespaceLiterals();
public void AppendSubBuilder(
   System.Web.UI.ControlBuilder subBuilder
);
```

MonthChangedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MonthChangedEventArgs
   extends System.Object
```

Public Constructors

```
public MonthChangedEventArgs(
   System.DateTime newDate,
   System.DateTime previousDate
);
```

Public Properties

```
public System.DateTime get_NewDate();
public System.DateTime get_PreviousDate();
```

MonthChangedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void MonthChangedEventHandler(
   System.Object sender,
   System.Web.UI.WebControls.MonthChangedEventArgs e
);
```

NextPrevFormat Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum NextPrevFormat
```

Members

CustomText

FullMonth

ShortMonth

PagedDataSource Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PagedDataSource
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable, System.Componen
tModel.ITypedList
```

Public Constructors

```
public PagedDataSource();
```

Public Properties

```
public boolean get_AllowCustomPaging();
public void set_AllowCustomPaging(
   boolean value
);
public boolean get_AllowPaging();
public void set_AllowPaging(
   boolean value
);
public final int get_Count();
public int get_CurrentPageIndex();
public void set_CurrentPageIndex(
   int value
);
public System.Collections.IEnumerable get_DataSource();
public void set_DataSource(
   System.Collections.IEnumerable value
);
public int get_DataSourceCount();
public int get_FirstIndexInPage();
public boolean get_IsCustomPagingEnabled();
public boolean get_IsFirstPage();
public boolean get_IsLastPage();
public boolean get_IsPagingEnabled();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public int get_PageCount();
public int get_PageSize();
public void set_PageSize(
   int value
);
public final System.Object get_SyncRoot();
public int get_VirtualCount();
public void set_VirtualCount(
   int value
);
```

Public Methods

```
public final void CopyTo(
   System.Array array,
   int index
);
```

```
public final System.Collections.IEnumerator GetEnumerator();
public final System.ComponentModel.PropertyDescriptorCollection GetItemProperties(
   System.ComponentModel.PropertyDescriptor[] listAccessors
);
public final System.String GetListName(
   System.ComponentModel.PropertyDescriptor[] listAccessors
);
```

PagerMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PagerMode
```

Members

NextPrev

NumericPages

PagerPosition Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PagerPosition
```

Members

Bottom

Top

TopAndBottom

Panel Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.PanelDesigner
, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.Web.UI.ToolboxDataAttribute(Data = "<{0}:Panel runat=server>Panel</{0}:
Panel>") */
/** @attribute System.Web.UI.ParseChildrenAttribute(ChildrenAsProperties = false, DefaultProp
erty = "") */
/** @attribute System.Web.UI.PersistChildrenAttribute(Persist = true) */
public class Panel
   extends System.Web.UI.WebControls.WebControl
```

Public Constructors

```
public Panel();
```

Public Properties

```
public System.String get_BackImageUrl();
public void set_BackImageUrl(
   System.String value
);
public System.Web.UI.WebControls.HorizontalAlign get_HorizontalAlign();
public void set_HorizontalAlign(
   System.Web.UI.WebControls.HorizontalAlign value
);
public boolean get_Wrap();
public void set_Wrap(
   boolean value
);
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
```

PlaceHolder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ControlBuilderAttribute(BuilderType = System.Web.UI.WebControls.
PlaceHolderControlBuilder) */
public class PlaceHolder
   extends System.Web.UI.Control
```

Public Constructors

```
public PlaceHolder();
```

PlaceHolderControlBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PlaceHolderControlBuilder
   extends System.Web.UI.ControlBuilder
```

Public Constructors

```
public PlaceHolderControlBuilder();
```

Public Methods

```
public boolean AllowWhitespaceLiterals();
```

RadioButton Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.CheckBoxDesig
ner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class RadioButton
   extends System.Web.UI.WebControls.CheckBox
```

Public Constructors

```
public RadioButton();
```

Public Properties

```
public System.String get_GroupName();
public void set_GroupName(
   System.String value
);
```

Protected Methods

```
protected void OnPreRender(
   System.EventArgs e
);
```

RadioButtonList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ValidationPropertyAttribute(Name = "SelectedItem") */
public class RadioButtonList
   extends System.Web.UI.WebControls.ListControl
   implements System.Web.UI.WebControls.IRepeatInfoUser, System.Web.UI.INamingContainer, Syst
em.Web.UI.IPostBackDataHandler
```

Public Constructors

```
public RadioButtonList();
```

Public Properties

```
public int get_CellPadding();
public void set_CellPadding(
   int value
);
public int get_CellSpacing();
public void set_CellSpacing(
   int value
);
public int get_RepeatColumns();
public void set_RepeatColumns(
   int value
);
public System.Web.UI.WebControls.RepeatDirection get_RepeatDirection();
public void set_RepeatDirection(
   System.Web.UI.WebControls.RepeatDirection value
);
public System.Web.UI.WebControls.RepeatLayout get_RepeatLayout();
public void set_RepeatLayout(
   System.Web.UI.WebControls.RepeatLayout value
);
public System.Web.UI.WebControls.TextAlign get_TextAlign();
public void set_TextAlign(
   System.Web.UI.WebControls.TextAlign value
);
```

Protected Methods

```
protected System.Web.UI.WebControls.Style CreateControlStyle();
protected void Render(
   System.Web.UI.HtmlTextWriter writer
);
```

RangeValidator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ToolboxDataAttribute(Data = "<{0}:RangeValidator runat=server Er
rorMessage="RangeValidator"></{0}:RangeValidator>") */
public class RangeValidator
   extends System.Web.UI.WebControls.BaseCompareValidator
```

Public Constructors

```
public RangeValidator();
```

Public Properties

```
public System.String get_MaximumValue();
public void set_MaximumValue(
   System.String value
);
public System.String get_MinimumValue();
public void set_MinimumValue(
   System.String value
);
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected boolean ControlPropertiesValid();
protected boolean EvaluateIsValid();
```

RegularExpressionValidator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ToolboxDataAttribute(Data = "<{0}:RegularExpressionValidator run
at=server ErrorMessage="RegularExpressionValidator"></{0}:RegularExpressionValidator>") */
public class RegularExpressionValidator
   extends System.Web.UI.WebControls.BaseValidator
```

Public Constructors

```
public RegularExpressionValidator();
```

Public Properties

```
public System.String get_ValidationExpression();
public void set_ValidationExpression(
   System.String value
);
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected boolean EvaluateIsValid();
```

RepeatDirection Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum RepeatDirection
```

Members

Horizontal

Vertical

Repeater Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.PersistChildrenAttribute(Persist = false) */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ItemCommand") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "DataSource") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.RepeaterDesig
ner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.Web.UI.ParseChildrenAttribute(ChildrenAsProperties = true, DefaultProp
erty = "") */
public class Repeater
   extends System.Web.UI.Control
   implements System.Web.UI.INamingContainer
```

Public Constructors

```
public Repeater();
```

Public Properties

```
public System.Web.UI.ITemplate get_AlternatingItemTemplate();
public void set_AlternatingItemTemplate(
   System.Web.UI.ITemplate value
);
public System.Web.UI.ControlCollection get_Controls();
public System.String get_DataMember();
public void set_DataMember(
   System.String value
);
public System.Object get_DataSource();
public void set_DataSource(
   System.Object value
);
public System.Web.UI.ITemplate get_FooterTemplate();
public void set_FooterTemplate(
   System.Web.UI.ITemplate value
);
public System.Web.UI.ITemplate get_HeaderTemplate();
public void set_HeaderTemplate(
   System.Web.UI.ITemplate value
);
public System.Web.UI.WebControls.RepeaterItemCollection get_Items();
public System.Web.UI.ITemplate get_ItemTemplate();
public void set_ItemTemplate(
   System.Web.UI.ITemplate value
);
public System.Web.UI.ITemplate get_SeparatorTemplate();
public void set_SeparatorTemplate(
   System.Web.UI.ITemplate value
);
```

Public Methods

```
public void DataBind();
```

Public Events

```
public void add_ItemCommand(
   System.Web.UI.WebControls.RepeaterCommandEventHandler value
);
public void remove_ItemCommand(
   System.Web.UI.WebControls.RepeaterCommandEventHandler value
);
public void add_ItemCreated(
   System.Web.UI.WebControls.RepeaterItemEventHandler value
);
public void remove_ItemCreated(
   System.Web.UI.WebControls.RepeaterItemEventHandler value
);
public void add_ItemDataBound(
   System.Web.UI.WebControls.RepeaterItemEventHandler value
);
public void remove_ItemDataBound(
   System.Web.UI.WebControls.RepeaterItemEventHandler value
);
```

Protected Methods

```
protected void CreateChildControls();
protected void CreateControlHierarchy(
   boolean useDataSource
);
protected System.Web.UI.WebControls.RepeaterItem CreateItem(
   int itemIndex,
   System.Web.UI.WebControls.ListItemType itemType
);
protected void InitializeItem(
   System.Web.UI.WebControls.RepeaterItem item
);
protected boolean OnBubbleEvent(
   System.Object sender,
   System.EventArgs e
);
protected void OnDataBinding(
   System.EventArgs e
);
protected void OnItemCommand(
   System.Web.UI.WebControls.RepeaterCommandEventArgs e
);
protected void OnItemCreated(
   System.Web.UI.WebControls.RepeaterItemEventArgs e
);
protected void OnItemDataBound(
   System.Web.UI.WebControls.RepeaterItemEventArgs e
);
```

RepeaterCommandEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RepeaterCommandEventArgs
   extends System.Web.UI.WebControls.CommandEventArgs
```

Public Constructors

```
public RepeaterCommandEventArgs(
   System.Web.UI.WebControls.RepeaterItem item,
   System.Object commandSource,
   System.Web.UI.WebControls.CommandEventArgs originalArgs
);
```

Public Properties

```
public System.Object get_CommandSource();
public System.Web.UI.WebControls.RepeaterItem get_Item();
```

RepeaterCommandEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void RepeaterCommandEventHandler(
   System.Object source,
   System.Web.UI.WebControls.RepeaterCommandEventArgs e
);
```

RepeaterItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class RepeaterItem
   extends System.Web.UI.Control
   implements System.Web.UI.INamingContainer
```

Public Constructors

```
public RepeaterItem(
   int itemIndex,
   System.Web.UI.WebControls.ListItemType itemType
);
```

Public Properties

```
public System.Object get_DataItem();
public void set_DataItem(
   System.Object value
);
public int get_ItemIndex();
public System.Web.UI.WebControls.ListItemType get_ItemType();
```

Protected Methods

```
protected boolean OnBubbleEvent(
   System.Object source,
   System.EventArgs e
);
```

RepeaterItemCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class RepeaterItemCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public RepeaterItemCollection(
   System.Collections.ArrayList items
);
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Web.UI.WebControls.RepeaterItem get_Item(
   int index
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
```

RepeaterItemEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RepeaterItemEventArgs
   extends System.EventArgs
```

Public Constructors

```
public RepeaterItemEventArgs(
   System.Web.UI.WebControls.RepeaterItem item
);
```

Public Properties

```
public System.Web.UI.WebControls.RepeaterItem get_Item();
```

RepeaterItemEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void RepeaterItemEventHandler(
   System.Object sender,
   System.Web.UI.WebControls.RepeaterItemEventArgs e
);
```

RepeatInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RepeatInfo
   extends System.Object
```

Public Constructors

```
public RepeatInfo();
```

Public Properties

```
public boolean get_OuterTableImplied();
public void set_OuterTableImplied(
   boolean value
);
public int get_RepeatColumns();
public void set_RepeatColumns(
   int value
);
public System.Web.UI.WebControls.RepeatDirection get_RepeatDirection();
public void set_RepeatDirection(
   System.Web.UI.WebControls.RepeatDirection value
);
public System.Web.UI.WebControls.RepeatLayout get_RepeatLayout();
public void set_RepeatLayout(
   System.Web.UI.WebControls.RepeatLayout value
);
```

Public Methods

```
public void RenderRepeater(
   System.Web.UI.HtmlTextWriter writer,
   System.Web.UI.WebControls.IRepeatInfoUser user,
   System.Web.UI.WebControls.Style controlStyle,
   System.Web.UI.WebControls.WebControl baseControl
);
```

RepeatLayout Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum RepeatLayout
```

Members

Flow

Table

RequiredFieldValidator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ToolboxDataAttribute(Data = "<{0}:RequiredFieldValidator runat=s
erver ErrorMessage="RequiredFieldValidator"></{0}:RequiredFieldValidator>") */
public class RequiredFieldValidator
   extends System.Web.UI.WebControls.BaseValidator
```

Public Constructors

```
public RequiredFieldValidator();
```

Public Properties

```
public System.String get_InitialValue();
public void set_InitialValue(
   System.String value
);
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected boolean EvaluateIsValid();
```

SelectedDatesCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SelectedDatesCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public SelectedDatesCollection(
   System.Collections.ArrayList dateList
);
```

Public Properties

```
public final int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.DateTime get_Item(
   int index
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public void Add(
   System.DateTime date
);
public void Clear();
public boolean Contains(
   System.DateTime date
);
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public void Remove(
   System.DateTime date
);
public void SelectRange(
   System.DateTime fromDate,
   System.DateTime toDate
);
```

ServerValidateEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ServerValidateEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ServerValidateEventArgs(
   System.String value,
   boolean isValid
);
```

Public Properties

```
public boolean get_IsValid();
public void set_IsValid(
   boolean value
);
public System.String get_Value();
```

ServerValidateEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ServerValidateEventHandler(
   System.Object source,
   System.Web.UI.WebControls.ServerValidateEventArgs args
);
```

Style Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Compo
nentModel.ExpandableObjectConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyTo
ken=b77a5c561934e089") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class Style
   extends System.ComponentModel.Component
   implements System.Web.UI.IStateManager
```

Public Constructors

```
public Style();
public Style(
   System.Web.UI.StateBag bag
);
```

Public Properties

```
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_BorderColor();
public void set_BorderColor(
   System.Drawing.Color value
);
public System.Web.UI.WebControls.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Web.UI.WebControls.BorderStyle value
);
public System.Web.UI.WebControls.Unit get_BorderWidth();
public void set_BorderWidth(
   System.Web.UI.WebControls.Unit value
);
public System.String get_CssClass();
public void set_CssClass(
   System.String value
);
public System.Web.UI.WebControls.FontInfo get_Font();
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.Web.UI.WebControls.Unit get_Height();
public void set_Height(
   System.Web.UI.WebControls.Unit value
);
public System.Web.UI.WebControls.Unit get_Width();
public void set_Width(
   System.Web.UI.WebControls.Unit value
);
```

Public Methods

```
public void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
public void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer,
   System.Web.UI.WebControls.WebControl owner
);
public void CopyFrom(
   System.Web.UI.WebControls.Style s
);
public void MergeWith(
   System.Web.UI.WebControls.Style s
);
public void Reset();
public System.String ToString();
```

Protected Properties

```
protected boolean get_IsEmpty();
protected boolean get_IsTrackingViewState();
protected System.Web.UI.StateBag get_ViewState();
```

Protected Methods

```
protected void LoadViewState(
   System.Object state
);
protected System.Object SaveViewState();
protected void SetBit(
   int bit
);
protected void TrackViewState();
```

Table Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.ParseChildrenAttribute(ChildrenAsProperties = true, DefaultProperty = "Rows") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Rows") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.ComponentModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.TableDesigner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class Table
   extends System.Web.UI.WebControls.WebControl
```

Public Constructors

```
public Table();
```

Public Properties

```
public System.String get_BackImageUrl();
public void set_BackImageUrl(
   System.String value
);
public int get_CellPadding();
public void set_CellPadding(
   int value
);
public int get_CellSpacing();
public void set_CellSpacing(
   int value
);
public System.Web.UI.WebControls.GridLines get_GridLines();
public void set_GridLines(
   System.Web.UI.WebControls.GridLines value
);
public System.Web.UI.WebControls.HorizontalAlign get_HorizontalAlign();
public void set_HorizontalAlign(
   System.Web.UI.WebControls.HorizontalAlign value
);
public System.Web.UI.WebControls.TableRowCollection get_Rows();
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected System.Web.UI.ControlCollection CreateControlCollection();
protected System.Web.UI.WebControls.Style CreateControlStyle();
protected void RenderContents(
   System.Web.UI.HtmlTextWriter writer
);
```

Table.RowControlCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
protected class RowControlCollection
   extends System.Web.UI.ControlCollection
```

Public Methods

```
public void Add(
   System.Web.UI.Control child
);
public void AddAt(
   int index,
   System.Web.UI.Control child
);
```

TableCell Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.Web.UI.ControlBuilderAttribute(BuilderType = System.Web.UI.WebControls.
TableCellControlBuilder) */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.Web.UI.ParseChildrenAttribute(ChildrenAsProperties = false, DefaultProp
erty = "") */
/** @attribute System.Web.UI.PersistChildrenAttribute(Persist = true) */
public class TableCell
   extends System.Web.UI.WebControls.WebControl
```

Public Constructors

```
public TableCell();
```

Public Properties

```
public int get_ColumnSpan();
public void set_ColumnSpan(
   int value
);
public System.Web.UI.WebControls.HorizontalAlign get_HorizontalAlign();
public void set_HorizontalAlign(
   System.Web.UI.WebControls.HorizontalAlign value
);
public int get_RowSpan();
public void set_RowSpan(
   int value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.Web.UI.WebControls.VerticalAlign get_VerticalAlign();
public void set_VerticalAlign(
   System.Web.UI.WebControls.VerticalAlign value
);
public boolean get_Wrap();
public void set_Wrap(
   boolean value
);
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected void AddParsedSubObject(
   System.Object obj
);
protected System.Web.UI.WebControls.Style CreateControlStyle();
protected void RenderContents(
   System.Web.UI.HtmlTextWriter writer
);
```

TableCellCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Web.UI.Design.WebControls.TableCellsCollectionEditor, Sy
stem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class TableCellCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Web.UI.WebControls.TableCell get_Item(
   int index
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public int Add(
   System.Web.UI.WebControls.TableCell cell
);
public void AddAt(
   int index,
   System.Web.UI.WebControls.TableCell cell
);
public void AddRange(
   System.Web.UI.WebControls.TableCell[] cells
);
public final void Clear();
public final void CopyTo(
   System.Array array,
   int index
);
public int GetCellIndex(
   System.Web.UI.WebControls.TableCell cell
);
public final System.Collections.IEnumerator GetEnumerator();
public void Remove(
   System.Web.UI.WebControls.TableCell cell
);
public final void RemoveAt(
   int index
);
```

TableCellControlBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TableCellControlBuilder
   extends System.Web.UI.ControlBuilder
```

Public Constructors

```
public TableCellControlBuilder();
```

Public Methods

```
public boolean AllowWhitespaceLiterals();
```

TableHeaderCell Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TableHeaderCell
   extends System.Web.UI.WebControls.TableCell
```

Public Constructors

```
public TableHeaderCell();
```

TableItemStyle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TableItemStyle
   extends System.Web.UI.WebControls.Style
```

Public Constructors

```
public TableItemStyle();
public TableItemStyle(
   System.Web.UI.StateBag bag
);
```

Public Properties

```
public System.Web.UI.WebControls.HorizontalAlign get_HorizontalAlign();
public void set_HorizontalAlign(
   System.Web.UI.WebControls.HorizontalAlign value
);
public System.Web.UI.WebControls.VerticalAlign get_VerticalAlign();
public void set_VerticalAlign(
   System.Web.UI.WebControls.VerticalAlign value
);
public boolean get_Wrap();
public void set_Wrap(
   boolean value
);
```

Public Methods

```
public void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer,
   System.Web.UI.WebControls.WebControl owner
);
public void CopyFrom(
   System.Web.UI.WebControls.Style s
);
public void MergeWith(
   System.Web.UI.WebControls.Style s
);
public void Reset();
```

TableRow Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Cells") */
/** @attribute System.Web.UI.ParseChildrenAttribute(ChildrenAsProperties = true, DefaultProperty = "Cells") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class TableRow
   extends System.Web.UI.WebControls.WebControl
```

Public Constructors

```
public TableRow();
```

Public Properties

```
public System.Web.UI.WebControls.TableCellCollection get_Cells();
public System.Web.UI.WebControls.HorizontalAlign get_HorizontalAlign();
public void set_HorizontalAlign(
   System.Web.UI.WebControls.HorizontalAlign value
);
public System.Web.UI.WebControls.VerticalAlign get_VerticalAlign();
public void set_VerticalAlign(
   System.Web.UI.WebControls.VerticalAlign value
);
```

Protected Methods

```
protected System.Web.UI.ControlCollection CreateControlCollection();
protected System.Web.UI.WebControls.Style CreateControlStyle();
```

TableRow.CellControlCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
protected class CellControlCollection
   extends System.Web.UI.ControlCollection
```

Public Methods

```
public void Add(
   System.Web.UI.Control child
);
public void AddAt(
   int index,
   System.Web.UI.Control child
);
```

TableRowCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Web.UI.Design.WebControls.TableRowsCollectionEditor, Sys
tem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class TableRowCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public System.Web.UI.WebControls.TableRow get_Item(
   int index
);
public final System.Object get_SyncRoot();
```

Public Methods

```
public int Add(
   System.Web.UI.WebControls.TableRow row
);
public void AddAt(
   int index,
   System.Web.UI.WebControls.TableRow row
);
public void AddRange(
   System.Web.UI.WebControls.TableRow[] rows
);
public final void Clear();
public final void CopyTo(
   System.Array array,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public int GetRowIndex(
   System.Web.UI.WebControls.TableRow row
);
public void Remove(
   System.Web.UI.WebControls.TableRow row
);
public final void RemoveAt(
   int index
);
```

TableStyle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TableStyle
   extends System.Web.UI.WebControls.Style
```

Public Constructors

```
public TableStyle();
public TableStyle(
   System.Web.UI.StateBag bag
);
```

Public Properties

```
public System.String get_BackImageUrl();
public void set_BackImageUrl(
   System.String value
);
public int get_CellPadding();
public void set_CellPadding(
   int value
);
public int get_CellSpacing();
public void set_CellSpacing(
   int value
);
public System.Web.UI.WebControls.GridLines get_GridLines();
public void set_GridLines(
   System.Web.UI.WebControls.GridLines value
);
public System.Web.UI.WebControls.HorizontalAlign get_HorizontalAlign();
public void set_HorizontalAlign(
   System.Web.UI.WebControls.HorizontalAlign value
);
```

Public Methods

```
public void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer,
   System.Web.UI.WebControls.WebControl owner
);
public void CopyFrom(
   System.Web.UI.WebControls.Style s
);
public void MergeWith(
   System.Web.UI.WebControls.Style s
);
public void Reset();
```

TargetConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TargetConverter
   extends System.ComponentModel.StringConverter
```

Public Constructors

```
public TargetConverter();
```

Public Methods

```
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

TemplateColumn Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TemplateColumn
   extends System.Web.UI.WebControls.DataGridColumn
```

Public Constructors

```
public TemplateColumn();
```

Public Properties

```
public System.Web.UI.ITemplate get_EditItemTemplate();
public void set_EditItemTemplate(
   System.Web.UI.ITemplate value
);
public System.Web.UI.ITemplate get_FooterTemplate();
public void set_FooterTemplate(
   System.Web.UI.ITemplate value
);
public System.Web.UI.ITemplate get_HeaderTemplate();
public void set_HeaderTemplate(
   System.Web.UI.ITemplate value
);
public System.Web.UI.ITemplate get_ItemTemplate();
public void set_ItemTemplate(
   System.Web.UI.ITemplate value
);
```

Public Methods

```
public void InitializeCell(
   System.Web.UI.WebControls.TableCell cell,
   int columnIndex,
   System.Web.UI.WebControls.ListItemType itemType
);
```

TextAlign Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TextAlign
```

Members

Left

Right

TextBox Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "TextChanged") */
/** @attribute System.Web.UI.ControlBuilderAttribute(BuilderType = System.Web.UI.WebControls.
TextBoxControlBuilder) */
/** @attribute System.Web.UI.DataBindingHandlerAttribute(HandlerTypeName = "System.Web.UI.Des
ign.TextDataBindingHandler, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToke
n=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.Web.UI.ValidationPropertyAttribute(Name = "Text") */
/** @attribute System.Web.UI.ParseChildrenAttribute(ChildrenAsProperties = false, DefaultProp
erty = "") */
public class TextBox
   extends System.Web.UI.WebControls.WebControl
   implements System.Web.UI.IPostBackDataHandler
```

Public Constructors

```
public TextBox();
```

Public Properties

```
public boolean get_AutoPostBack();
public void set_AutoPostBack(
   boolean value
);
public int get_Columns();
public void set_Columns(
   int value
);
public int get_MaxLength();
public void set_MaxLength(
   int value
);
public boolean get_ReadOnly();
public void set_ReadOnly(
   boolean value
);
public int get_Rows();
public void set_Rows(
   int value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.Web.UI.WebControls.TextBoxMode get_TextMode();
public void set_TextMode(
   System.Web.UI.WebControls.TextBoxMode value
);
public boolean get_Wrap();
public void set_Wrap(
   boolean value
);
```

Public Events

```
public void add_TextChanged(
   System.EventHandler value
);
public void remove_TextChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Web.UI.HtmlTextWriterTag get_TagKey();
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected void AddParsedSubObject(
   System.Object obj
);
protected void OnPreRender(
   System.EventArgs e
);
protected void OnTextChanged(
   System.EventArgs e
);
protected void Render(
   System.Web.UI.HtmlTextWriter writer
);
```

TextBoxControlBuilder Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TextBoxControlBuilder
   extends System.Web.UI.ControlBuilder
```

Public Constructors

```
public TextBoxControlBuilder();
```

Public Methods

```
public boolean AllowWhitespaceLiterals();
public boolean HtmlDecodeLiterals();
```

TextBoxMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TextBoxMode
```

Members

MultiLine

Password

SingleLine

TitleFormat Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TitleFormat
```

Members

Month

MonthYear

Unit Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Web.U
I.WebControls.UnitConverter, System.Web, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=
b03f5f7f11d50a3a") */
public class Unit
   extends System.ValueType
```

Public Constructors

```
public Unit(
   double value
);
public Unit(
   int value
);
public Unit(
   System.String value
);
public Unit(
   double value,
   System.Web.UI.WebControls.UnitType type
);
public Unit(
   System.String value,
   System.Globalization.CultureInfo culture
);
```

Public Properties

```
public boolean get_IsEmpty();
public System.Web.UI.WebControls.UnitType get_Type();
public double get_Value();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public static System.Web.UI.WebControls.Unit Parse(
   System.String s
);
public static System.Web.UI.WebControls.Unit Parse(
   System.String s,
   System.Globalization.CultureInfo culture
);
public static System.Web.UI.WebControls.Unit Percentage(
   double n
);
public static System.Web.UI.WebControls.Unit Pixel(
   int n
);
public static System.Web.UI.WebControls.Unit Point(
   int n
```

```
);
public System.String ToString();
public System.String ToString(
   System.Globalization.CultureInfo culture
);
```

Public Operators

```
public static boolean op_Equality(
   System.Web.UI.WebControls.Unit left,
   System.Web.UI.WebControls.Unit right
);
public static System.Web.UI.WebControls.Unit op_Implicit(
   int n
);
public static boolean op_Inequality(
   System.Web.UI.WebControls.Unit left,
   System.Web.UI.WebControls.Unit right
);
```

Public Fields

```
public static final System.Web.UI.WebControls.Unit Empty;
```

UnitConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UnitConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public UnitConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

UnitType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum UnitType
```

Members

Cm

Em

Ex

Inch

Mm

Percentage

Pica

Pixel

Point

ValidatedControlConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ValidatedControlConverter
   extends System.ComponentModel.StringConverter
```

Public Constructors

```
public ValidatedControlConverter();
```

Public Methods

```
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

ValidationCompareOperator Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ValidationCompareOperator
```

Members

DataTypeCheck

Equal

GreaterThan

GreaterThanEqual

LessThan

LessThanEqual

NotEqual

ValidationDataType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ValidationDataType
```

Members

Currency

Date

Double

Integer

String

ValidationSummary Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ValidationSummary
   extends System.Web.UI.WebControls.WebControl
```

Public Constructors

```
public ValidationSummary();
```

Public Properties

```
public System.Web.UI.WebControls.ValidationSummaryDisplayMode get_DisplayMode();
public void set_DisplayMode(
   System.Web.UI.WebControls.ValidationSummaryDisplayMode value
);
public boolean get_EnableClientScript();
public void set_EnableClientScript(
   boolean value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.String get_HeaderText();
public void set_HeaderText(
   System.String value
);
public boolean get_ShowMessageBox();
public void set_ShowMessageBox(
   boolean value
);
public boolean get_ShowSummary();
public void set_ShowSummary(
   boolean value
);
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected void OnPreRender(
   System.EventArgs e
);
protected void Render(
   System.Web.UI.HtmlTextWriter writer
);
```

ValidationSummaryDisplayMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ValidationSummaryDisplayMode
```

Members

BulletList

List

SingleParagraph

ValidatorDisplay Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ValidatorDisplay
```

Members

Dynamic

None

Static

VerticalAlign Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Web.U
I.WebControls.VerticalAlignConverter, System.Web, Version=1.0.3300.0, Culture=neutral, Public
KeyToken=b03f5f7f11d50a3a") */
public enum VerticalAlign
```

Members

Bottom

Middle

NotSet

Top

WebColorConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class WebColorConverter
   extends System.Drawing.ColorConverter
```

Public Constructors

```
public WebColorConverter();
```

Public Methods

```
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

WebControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.PersistChildrenAttribute(Persist = false) */
/** @attribute System.Web.UI.ParseChildrenAttribute(ChildrenAsProperties = true, DefaultProperty = "") */
public class WebControl
   extends System.Web.UI.Control
   implements System.Web.UI.IAttributeAccessor
```

Public Constructors

```
public WebControl(
   System.Web.UI.HtmlTextWriterTag tag
);
```

Public Properties

```
public System.String get_AccessKey();
public void set_AccessKey(
   System.String value
);
public System.Web.UI.AttributeCollection get_Attributes();
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_BorderColor();
public void set_BorderColor(
   System.Drawing.Color value
);
public System.Web.UI.WebControls.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Web.UI.WebControls.BorderStyle value
);
public System.Web.UI.WebControls.Unit get_BorderWidth();
public void set_BorderWidth(
   System.Web.UI.WebControls.Unit value
);
public System.Web.UI.WebControls.Style get_ControlStyle();
public boolean get_ControlStyleCreated();
public System.String get_CssClass();
public void set_CssClass(
   System.String value
);
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public System.Web.UI.WebControls.FontInfo get_Font();
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.Web.UI.WebControls.Unit get_Height();
public void set_Height(
   System.Web.UI.WebControls.Unit value
);
```

```
public System.Web.UI.CssStyleCollection get_Style();
public short get_TabIndex();
public void set_TabIndex(
   short value
);
public System.String get_ToolTip();
public void set_ToolTip(
   System.String value
);
public System.Web.UI.WebControls.Unit get_Width();
public void set_Width(
   System.Web.UI.WebControls.Unit value
);
```

Public Methods

```
public void ApplyStyle(
   System.Web.UI.WebControls.Style s
);
public void CopyBaseAttributes(
   System.Web.UI.WebControls.WebControl controlSrc
);
public void MergeStyle(
   System.Web.UI.WebControls.Style s
);
public void RenderBeginTag(
   System.Web.UI.HtmlTextWriter writer
);
public void RenderEndTag(
   System.Web.UI.HtmlTextWriter writer
);
```

Protected Constructors

```
protected WebControl();
protected WebControl(
   System.String tag
);
```

Protected Properties

```
protected System.Web.UI.HtmlTextWriterTag get_TagKey();
protected System.String get_TagName();
```

Protected Methods

```
protected void AddAttributesToRender(
   System.Web.UI.HtmlTextWriter writer
);
protected System.Web.UI.WebControls.Style CreateControlStyle();
protected void LoadViewState(
   System.Object savedState
);
protected void Render(
   System.Web.UI.HtmlTextWriter writer
);
protected void RenderContents(
   System.Web.UI.HtmlTextWriter writer
);
protected System.Object SaveViewState();
```

```
protected void TrackViewState();
```

Xml Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Web.UI.PersistChildrenAttribute(Persist = false) */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "DocumentSource") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Web.UI.Design.WebControls.XmlDesigner,
System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.Web.UI.ControlBuilderAttribute(BuilderType = System.Web.UI.WebControls.
XmlBuilder) */
public class Xml
   extends System.Web.UI.Control
```

Public Constructors

```
public Xml();
```

Public Properties

```
public System.Xml.XmlDocument get_Document();
public void set_Document(
   System.Xml.XmlDocument value
);
public System.String get_DocumentContent();
public void set_DocumentContent(
   System.String value
);
public System.String get_DocumentSource();
public void set_DocumentSource(
   System.String value
);
public System.Xml.Xsl.XslTransform get_Transform();
public void set_Transform(
   System.Xml.Xsl.XslTransform value
);
public System.Xml.Xsl.XsltArgumentList get_TransformArgumentList();
public void set_TransformArgumentList(
   System.Xml.Xsl.XsltArgumentList value
);
public System.String get_TransformSource();
public void set_TransformSource(
   System.String value
);
```

Protected Methods

```
protected void AddParsedSubObject(
   System.Object obj
);
protected void Render(
   System.Web.UI.HtmlTextWriter output
);
```

System.Windows.Forms Namespace (Visual J# Syntax)

Classes

AccessibleObject

AmbientProperties

Application

ApplicationContext

AxHost

AxHost.AxComponentEditor

AxHost.ClsidAttribute

AxHost.ConnectionPointCookie

AxHost.InvalidActiveXStateException

AxHost.State

AxHost.StateConverter

AxHost.TypeLibraryTimeStampAttribute

BaseCollection

Binding

BindingContext

BindingManagerBase

BindingsCollection

Button

ButtonBase

ButtonBase.ButtonBaseAccessibleObject

CheckBox

CheckBox.CheckBoxAccessibleObject

CheckedListBox

CheckedListBox.CheckedIndexCollection

CheckedListBox.CheckedItemCollection

CheckedListBox.ObjectCollection

Clipboard

ColorDialog

ColumnClickEventArgs

ColumnHeader

ComboBox

ComboBox.ChildAccessibleObject

ComboBox.ObjectCollection

CommonDialog

ContainerControl

ContentsResizedEventArgs

ContextMenu

Control

Control.ControlAccessibleObject

Control.ControlCollection

ControlBindingsCollection

ControlEventArgs

ControlPaint

ConvertEventArgs

CreateParams

CurrencyManager

Cursor

CursorConverter

Cursors

DataFormats

DataFormats.Format

DataGrid

DataGrid.HitTestInfo

DataGridBoolColumn

DataGridColumnStyle

DataGridColumnStyle.CompModSwitches

DataGridColumnStyle.DataGridColumnHeaderAccessibleObject

DataGridPreferredColumnWidthTypeConverter

DataGridTableStyle

DataGridTextBox

DataGridTextBoxColumn

DataObject

DateBoldEventArgs

DateRangeEventArgs

DateTimePicker

DateTimePicker.DateTimePickerAccessibleObject

DomainUpDown

DomainUpDown.DomainItemAccessibleObject

DomainUpDown.DomainUpDownAccessibleObject

DomainUpDown.DomainUpDownItemCollection

DragEventArgs

DrawItemEventArgs

ErrorProvider

FeatureSupport

FileDialog

FontDialog

Form

Form.ControlCollection

GiveFeedbackEventArgs

GridColumnStylesCollection

GridItem

GridItemCollection

GridTablesFactory

GridTableStylesCollection

GroupBox

Help

HelpEventArgs

HelpProvider

HScrollBar

ImageIndexConverter

ImageList

ImageList.ImageCollection

ImageListStreamer

InputLanguage

InputLanguageChangedEventArgs

InputLanguageChangingEventArgs

InputLanguageCollection

InvalidateEventArgs

ItemChangedEventArgs

ItemCheckEventArgs

ItemDragEventArgs

KeyEventArgs

KeyPressEventArgs

KeysConverter

Label

LabelEditEventArgs

LayoutEventArgs

LinkArea.LinkAreaConverter

LinkClickedEventArgs

LinkLabel

LinkLabel.Link

LinkLabel.LinkCollection

LinkLabelLinkClickedEventArgs

ListBindingConverter

ListBox

ListBox.ObjectCollection

ListBox.SelectedIndexCollection

ListBox.SelectedObjectCollection

ListControl

ListView

ListView.CheckedIndexCollection

ListView.CheckedListViewItemCollection

ListView.ColumnHeaderCollection

ListView.ListViewItemCollection

ListView.SelectedIndexCollection

ListView.SelectedListViewItemCollection

ListViewItem

ListViewItem.ListViewSubItem

ListViewItem.ListViewSubItemCollection

ListViewItemConverter

MainMenu

MdiClient

MdiClient.ControlCollection

MeasureItemEventArgs

Menu

Menu.MenuItemCollection

MenuItem

MessageBox

MonthCalendar

MonthCalendar.HitTestInfo

MouseEventArgs

NativeWindow

NavigateEventArgs

NodeLabelEditEventArgs

NotifyIcon

NumericUpDown

OpacityConverter

OpenFileDialog

OSFeature

OwnerDrawPropertyBag

PageSetupDialog

PaintEventArgs

Panel

PictureBox

PrintControllerWithStatusDialog

PrintDialog

PrintPreviewControl

PrintPreviewDialog

ProgressBar

PropertyGrid

PropertyGrid.PropertyTabCollection

PropertyManager

PropertyTabChangedEventArgs

PropertyValueChangedEventArgs

QueryAccessibilityHelpEventArgs

QueryContinueDragEventArgs

RadioButton

RadioButton.RadioButtonAccessibleObject

RichTextBox

SaveFileDialog

Screen

ScrollableControl

ScrollableControl.DockPaddingEdges

ScrollableControl.DockPaddingEdgesConverter

ScrollBar

ScrollEventArgs

SelectedGridItemChangedEventArgs

SelectionRange

SelectionRangeConverter

SendKeys

Splitter

SplitterEventArgs

StatusBar

StatusBar.StatusBarPanelCollection

StatusBarDrawItemEventArgs

StatusBarPanel

StatusBarPanelClickEventArgs

SystemInformation

TabControl

TabControl.ControlCollection

TabControl.TabPageCollection

TabPage

TabPage.TabPageControlCollection

TextBox

TextBoxBase

ThreadExceptionDialog

Timer

ToolBar

ToolBar.ToolBarButtonCollection

ToolBarButton

ToolBarButtonClickEventArgs

ToolTip

TrackBar

TreeNode

TreeNodeCollection

TreeNodeConverter

TreeView

TreeViewCancelEventArgs

TreeViewEventArgs

TreeViewImageIndexConverter

UICuesEventArgs

UpDownBase

UpDownEventArgs

UserControl

VScrollBar

Interfaces

IButtonControl

ICommandExecutor

IComponentEditorPageSite

IContainerControl

IDataGridColumnStyleEditingNotificationService

IDataGridEditingService

IDataObject

IFeatureSupport

IFileReaderService

IMessageFilter

IWin32Window

IWindowTarget

Structures

BindingMemberInfo

DataGridCell

LinkArea

Message

Delegates

AxHost.AboutBoxDelegate

ColumnClickEventHandler

ContentsResizedEventHandler

ControlEventHandler

ConvertEventHandler

DateBoldEventHandler

DateRangeEventHandler

DragEventHandler

DrawItemEventHandler

GiveFeedbackEventHandler

HelpEventHandler

InputLanguageChangedEventHandler

InputLanguageChangingEventHandler

InvalidateEventHandler

ItemChangedEventHandler

ItemCheckEventHandler

ItemDragEventHandler

KeyEventHandler

KeyPressEventHandler

LabelEditEventHandler

LayoutEventHandler

LinkClickedEventHandler

LinkLabelLinkClickedEventHandler

MeasureItemEventHandler

MethodInvoker

MouseEventHandler

NavigateEventHandler

NodeLabelEditEventHandler

PaintEventHandler

PropertyTabChangedEventHandler

PropertyValueChangedEventHandler

QueryAccessibilityHelpEventHandler

QueryContinueDragEventHandler

ScrollEventHandler

SelectedGridItemChangedEventHandler

SplitterEventHandler

StatusBarDrawItemEventHandler

StatusBarPanelClickEventHandler

ToolBarButtonClickEventHandler

TreeViewCancelEventHandler

TreeViewEventHandler

UICuesEventHandler

UpDownEventHandler

Enumerations

AccessibleEvents

AccessibleNavigation

AccessibleRole

AccessibleSelection

AccessibleStates

AnchorStyles

Appearance

ArrangeDirection

ArrangeStartingPosition

AxHost.ActiveXInvokeKind

BootMode

Border3DSide

Border3DStyle

BorderStyle

BoundsSpecified

ButtonBorderStyle

ButtonState

CaptionButton

CharacterCasing

CheckState

ColorDepth

ColumnHeaderStyle

ComboBoxStyle

ControlStyles

DataGrid.HitTestType

DataGridLineStyle

DataGridParentRowsLabelStyle

DateTimePickerFormat

Day

DialogResult

DockStyle

DragAction

DragDropEffects

DrawItemState

DrawMode

ErrorBlinkStyle

ErrorIconAlignment

FlatStyle

FormBorderStyle

FormStartPosition

FormWindowState

FrameStyle

GridItemType

HelpNavigator

HorizontalAlignment

ImeMode

ItemActivation

ItemBoundsPortion

Keys

LeftRightAlignment

LinkBehavior

LinkState

ListViewAlignment

MdiLayout

MenuGlyph

MenuMerge

MessageBoxButtons

MessageBoxDefaultButton

MessageBoxIcon

MessageBoxOptions

MonthCalendar.HitArea

MouseButtons

Orientation

PictureBoxSizeMode

PropertySort

RichTextBoxFinds

RichTextBoxScrollBars

RichTextBoxSelectionAttribute

RichTextBoxSelectionTypes

RichTextBoxStreamType

RichTextBoxWordPunctuations

RightToLeft

ScrollBars

ScrollButton

ScrollEventType

SecurityIDType

SelectionMode

Shortcut

SizeGripStyle

SortOrder

StatusBarPanelAutoSize

StatusBarPanelBorderStyle

StatusBarPanelStyle

StructFormat

TabAlignment

TabAppearance

TabDrawMode

TabSizeMode

TickStyle

ToolBarAppearance

ToolBarButtonStyle

ToolBarTextAlign

TreeViewAction

UICues

View

AccessibleEvents Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum AccessibleEvents
```

Members

AcceleratorChange

Create

DefaultActionChange

DescriptionChange

Destroy

Focus

HelpChange

Hide

LocationChange

NameChange

ParentChange

Reorder

Selection

SelectionAdd

SelectionRemove

SelectionWithin

Show

StateChange

SystemAlert

SystemCaptureEnd

SystemCaptureStart

SystemContextHelpEnd

SystemContextHelpStart

SystemDialogEnd

SystemDialogStart

SystemDragDropEnd

SystemDragDropStart

SystemForeground

SystemMenuEnd

SystemMenuPopupEnd

SystemMenuPopupStart

SystemMenuStart

SystemMinimizeEnd

SystemMinimizeStart

SystemMoveSizeEnd

SystemMoveSizeStart

SystemScrollingEnd

SystemScrollingStart

SystemSound

SystemSwitchEnd

SystemSwitchStart

ValueChange

AccessibleNavigation Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum AccessibleNavigation
```

Members

Down

FirstChild

LastChild

Left

Next

Previous

Right

Up

AccessibleObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class AccessibleObject
   extends System.MarshalByRefObject
   implements System.Reflection.IReflect, Accessibility.IAccessible, System.Windows.Forms.Uns
afeNativeMethods+IEnumVariant
```

Public Constructors

```
public AccessibleObject();
```

Public Properties

```
public System.Drawing.Rectangle get_Bounds();
public System.String get_DefaultAction();
public System.String get_Description();
public System.String get_Help();
public System.String get_KeyboardShortcut();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Windows.Forms.AccessibleObject get_Parent();
public System.Windows.Forms.AccessibleRole get_Role();
public System.Windows.Forms.AccessibleStates get_State();
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Methods

```
public void DoDefaultAction();
public System.Windows.Forms.AccessibleObject GetChild(
   int index
);
public int GetChildCount();
public System.Windows.Forms.AccessibleObject GetFocused();
public int GetHelpTopic(
   System.String& fileName
);
public System.Windows.Forms.AccessibleObject GetSelected();
public System.Windows.Forms.AccessibleObject HitTest(
   int x,
   int y
);
public System.Windows.Forms.AccessibleObject Navigate(
   System.Windows.Forms.AccessibleNavigation navdir
);
public void Select(
   System.Windows.Forms.AccessibleSelection flags
);
```

Protected Methods

```
protected void UseStdAccessibleObjects(
   System.IntPtr handle
);
protected void UseStdAccessibleObjects(
   System.IntPtr handle,
   int objid
);
```

AccessibleRole Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum AccessibleRole
```

Members

Alert

Animation

Application

Border

ButtonDropDown

ButtonDropDownGrid

ButtonMenu

Caret

Cell

Character

Chart

CheckButton

Client

Clock

Column

ColumnHeader

ComboBox

Cursor

Default

Diagram

Dial

Dialog

Document

DropList

Equation

Graphic

Grip

Grouping

HelpBalloon

HotkeyField

Indicator

Link

List

ListItem

MenuBar

MenuItem

MenuPopup

None

Outline

OutlineItem

PageTab

PageTabList

Pane

ProgressBar

PropertyPage

PushButton

RadioButton

Row

RowHeader

ScrollBar

Separator

Slider

Sound

SpinButton

StaticText

StatusBar

Table

Text

TitleBar

ToolBar

ToolTip

WhiteSpace

Window

AccessibleSelection Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum AccessibleSelection
```

Members

AddSelection

ExtendSelection

None

RemoveSelection

TakeFocus

TakeSelection

AccessibleStates Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum AccessibleStates
```

Members

AlertHigh

AlertLow

AlertMedium

Animated

Busy

Checked

Collapsed

Default

Expanded

ExtSelectable

Floating

Focusable

Focused

HotTracked

Indeterminate

Invisible

Linked

Marqueed

Mixed

Moveable

MultiSelectable

None

Offscreen

Pressed

Protected

ReadOnly

Selectable

Selected

SelfVoicing

Sizeable

Traversed

Unavailable

Valid

AmbientProperties Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AmbientProperties
   extends System.Object
```

Public Constructors

```
public AmbientProperties();
```

Public Properties

```
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Windows.Forms.Cursor get_Cursor();
public void set_Cursor(
   System.Windows.Forms.Cursor value
);
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
```

AnchorStyles Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Windows.Forms.Design.AnchorEditor, System.Design, Versio
n=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.FlagsAttribute() */
public enum AnchorStyles
```

Members

Bottom

Left

None

Right

Top

Appearance Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum Appearance
```

Members

Button

Normal

Application Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Application
   extends System.Object
```

Public Properties

```
public static boolean get_AllowQuit();
public static System.String get_CommonAppDataPath();
public static Microsoft.Win32.RegistryKey get_CommonAppDataRegistry();
public static System.String get_CompanyName();
public static System.Globalization.CultureInfo get_CurrentCulture();
public static void set_CurrentCulture(
   System.Globalization.CultureInfo value
);
public static System.Windows.Forms.InputLanguage get_CurrentInputLanguage();
public static void set_CurrentInputLanguage(
   System.Windows.Forms.InputLanguage value
);
public static System.String get_ExecutablePath();
public static System.String get_LocalUserAppDataPath();
public static boolean get_MessageLoop();
public static System.String get_ProductName();
public static System.String get_ProductVersion();
public static System.String get_SafeTopLevelCaptionFormat();
public static void set_SafeTopLevelCaptionFormat(
   System.String value
);
public static System.String get_StartupPath();
public static System.String get_UserAppDataPath();
public static Microsoft.Win32.RegistryKey get_UserAppDataRegistry();
```

Public Methods

```
public static void AddMessageFilter(
   System.Windows.Forms.IMessageFilter value
);
public static void DoEvents();
public static void Exit();
public static void ExitThread();
public static System.Threading.ApartmentState OleRequired();
public static void OnThreadException(
   System.Exception t
);
public static void RemoveMessageFilter(
   System.Windows.Forms.IMessageFilter value
);
public static void Run();
public static void Run(
   System.Windows.Forms.ApplicationContext context
);
public static void Run(
   System.Windows.Forms.Form mainForm
);
```

Public Events

```
public static void add_ApplicationExit(
   System.EventHandler value
);
public static void remove_ApplicationExit(
   System.EventHandler value
);
public static void add_Idle(
   System.EventHandler value
);
public static void remove_Idle(
   System.EventHandler value
);
public static void add_ThreadException(
   System.Threading.ThreadExceptionEventHandler value
);
public static void remove_ThreadException(
   System.Threading.ThreadExceptionEventHandler value
);
public static void add_ThreadExit(
   System.EventHandler value
);
public static void remove_ThreadExit(
   System.EventHandler value
);
```

ApplicationContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ApplicationContext
   extends System.Object
```

Public Constructors

```
public ApplicationContext();
public ApplicationContext(
   System.Windows.Forms.Form mainForm
);
```

Public Properties

```
public System.Windows.Forms.Form get_MainForm();
public void set_MainForm(
   System.Windows.Forms.Form value
);
```

Public Methods

```
public void Dispose();
public void ExitThread();
```

Public Events

```
public void add_ThreadExit(
   System.EventHandler value
);
public void remove_ThreadExit(
   System.EventHandler value
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void ExitThreadCore();
protected void Finalize();
protected void OnMainFormClosed(
   System.Object sender,
   System.EventArgs e
);
```

ArrangeDirection Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum ArrangeDirection
```

Members

Down

Left

Right

Up

ArrangeStartingPosition Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ArrangeStartingPosition
```

Members

BottomLeft

BottomRight

Hide

TopLeft

TopRight

AxHost Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Enter") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.AxHostDesigner, Sy
stem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public abstract class AxHost
   extends System.Windows.Forms.Control
   implements System.ComponentModel.ISupportInitialize, System.ComponentModel.ICustomTypeDesc
riptor
```

Public Properties

```
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.ContainerControl get_ContainingControl();
public void set_ContainingControl(
   System.Windows.Forms.ContainerControl value
);
public System.Windows.Forms.ContextMenu get_ContextMenu();
public void set_ContextMenu(
   System.Windows.Forms.ContextMenu value
);
public System.Windows.Forms.Cursor get_Cursor();
public void set_Cursor(
   System.Windows.Forms.Cursor value
);
public boolean get_EditMode();
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public boolean get_HasAboutBox();
public System.Windows.Forms.AxHost+State get_OcxState();
public void set_OcxState(
   System.Windows.Forms.AxHost+State value
);
public boolean get_RightToLeft();
public void set_RightToLeft(
   boolean value
);
public void set_Site(
   System.ComponentModel.ISite value
```

```
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Methods

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public final void BeginInit();
public void DoVerb(
   int verb
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public final void EndInit();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public System.Object GetOcx();
public boolean HasPropertyPages();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public void InvokeEditMode();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public void MakeDirty();
public boolean PreProcessMessage(
   System.Windows.Forms.Message& msg
);
public void ShowAboutBox();
public void ShowPropertyPages();
public void ShowPropertyPages(
   System.Windows.Forms.Control control
);
```

Public Events

```
public void add_BackColorChanged(
   System.EventHandler value
);
public void remove_BackColorChanged(
   System.EventHandler value
);
public void add_BackgroundImageChanged(
   System.EventHandler value
);
public void remove_BackgroundImageChanged(
   System.EventHandler value
);
public void add_BindingContextChanged(
   System.EventHandler value
);
public void remove_BindingContextChanged(
   System.EventHandler value
);
public void add_ChangeUICues(
   System.Windows.Forms.UICuesEventHandler value
);
public void remove_ChangeUICues(
   System.Windows.Forms.UICuesEventHandler value
);
public void add_Click(
   System.EventHandler value
);
public void remove_Click(
   System.EventHandler value
);
public void add_ContextMenuChanged(
```

```
    System.EventHandler value
);
public void remove_ContextMenuChanged(
    System.EventHandler value
);
public void add_CursorChanged(
    System.EventHandler value
);
public void remove_CursorChanged(
    System.EventHandler value
);
public void add_DoubleClick(
    System.EventHandler value
);
public void remove_DoubleClick(
    System.EventHandler value
);
public void add_DragDrop(
    System.Windows.Forms.DragEventHandler value
);
public void remove_DragDrop(
    System.Windows.Forms.DragEventHandler value
);
public void add_DragEnter(
    System.Windows.Forms.DragEventHandler value
);
public void remove_DragEnter(
    System.Windows.Forms.DragEventHandler value
);
public void add_DragLeave(
    System.EventHandler value
);
public void remove_DragLeave(
    System.EventHandler value
);
public void add_DragOver(
    System.Windows.Forms.DragEventHandler value
);
public void remove_DragOver(
    System.Windows.Forms.DragEventHandler value
);
public void add_EnabledChanged(
    System.EventHandler value
);
public void remove_EnabledChanged(
    System.EventHandler value
);
public void add_FontChanged(
    System.EventHandler value
);
public void remove_FontChanged(
    System.EventHandler value
);
public void add_ForeColorChanged(
    System.EventHandler value
);
public void remove_ForeColorChanged(
    System.EventHandler value
);
public void add_GiveFeedback(
    System.Windows.Forms.GiveFeedbackEventHandler value
);
public void remove_GiveFeedback(
    System.Windows.Forms.GiveFeedbackEventHandler value
);
public void add_HelpRequested(
    System.Windows.Forms.HelpEventHandler value
);
```

```
public void remove_HelpRequested(
   System.Windows.Forms.HelpEventHandler value
);
public void add_ImeModeChanged(
   System.EventHandler value
);
public void remove_ImeModeChanged(
   System.EventHandler value
);
public void add_KeyDown(
   System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyDown(
   System.Windows.Forms.KeyEventHandler value
);
public void add_KeyPress(
   System.Windows.Forms.KeyPressEventHandler value
);
public void remove_KeyPress(
   System.Windows.Forms.KeyPressEventHandler value
);
public void add_KeyUp(
   System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyUp(
   System.Windows.Forms.KeyEventHandler value
);
public void add_Layout(
   System.Windows.Forms.LayoutEventHandler value
);
public void remove_Layout(
   System.Windows.Forms.LayoutEventHandler value
);
public void add_MouseDown(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseDown(
   System.Windows.Forms.MouseEventHandler value
);
public void add_MouseEnter(
   System.EventHandler value
);
public void remove_MouseEnter(
   System.EventHandler value
);
public void add_MouseHover(
   System.EventHandler value
);
public void remove_MouseHover(
   System.EventHandler value
);
public void add_MouseLeave(
   System.EventHandler value
);
public void remove_MouseLeave(
   System.EventHandler value
);
public void add_MouseMove(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseMove(
   System.Windows.Forms.MouseEventHandler value
);
public void add_MouseUp(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseUp(
   System.Windows.Forms.MouseEventHandler value
```

```
);
public void add_MouseWheel(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseWheel(
   System.Windows.Forms.MouseEventHandler value
);
public void add_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void add_QueryAccessibilityHelp(
   System.Windows.Forms.QueryAccessibilityHelpEventHandler value
);
public void remove_QueryAccessibilityHelp(
   System.Windows.Forms.QueryAccessibilityHelpEventHandler value
);
public void add_QueryContinueDrag(
   System.Windows.Forms.QueryContinueDragEventHandler value
);
public void remove_QueryContinueDrag(
   System.Windows.Forms.QueryContinueDragEventHandler value
);
public void add_RightToLeftChanged(
   System.EventHandler value
);
public void remove_RightToLeftChanged(
   System.EventHandler value
);
public void add_StyleChanged(
   System.EventHandler value
);
public void remove_StyleChanged(
   System.EventHandler value
);
public void add_TabIndexChanged(
   System.EventHandler value
);
public void remove_TabIndexChanged(
   System.EventHandler value
);
public void add_TabStopChanged(
   System.EventHandler value
);
public void remove_TabStopChanged(
   System.EventHandler value
);
public void add_TextChanged(
   System.EventHandler value
);
public void remove_TextChanged(
   System.EventHandler value
);
```

Protected Constructors

```
protected AxHost(
   System.String clsid
);
protected AxHost(
   System.String clsid,
   int flags
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void AttachInterfaces();
protected void CreateHandle();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void CreateSink();
protected void DestroyHandle();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void DetachSink();
protected void Dispose(
   boolean disposing
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected static System.Drawing.Font GetFontFromIFont(
   System.Object font
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected static System.Drawing.Font GetFontFromIFontDisp(
   System.Object font
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected static System.Object GetIFontDispFromFont(
   System.Drawing.Font font
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected static System.Object GetIFontFromFont(
   System.Drawing.Font font
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected static System.Object GetIPictureDispFromPicture(
   System.Drawing.Image image
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected static System.Object GetIPictureFromCursor(
   System.Windows.Forms.Cursor cursor
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected static System.Object GetIPictureFromPicture(
   System.Drawing.Image image
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected static double GetOADateFromTime(
   System.DateTime time
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected static System.Drawing.Image GetPictureFromIPicture(
   System.Object picture
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected static System.Drawing.Image GetPictureFromIPictureDisp(
   System.Object picture
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected static System.DateTime GetTimeFromOADate(
   double date
);
protected boolean IsInputChar(
   char charCode
);
protected void OnBackColorChanged(
```

```
   System.EventArgs e
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnForeColorChanged(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnInPlaceActive();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnLostFocus(
   System.EventArgs e
);
protected boolean ProcessMnemonic(
   char charCode
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected boolean PropsValid();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RaiseOnMouseDown(
   short button,
   short shift,
   int x,
   int y
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RaiseOnMouseDown(
   short button,
   short shift,
   float x,
   float y
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RaiseOnMouseDown(
   System.Object o1,
   System.Object o2,
   System.Object o3,
   System.Object o4
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RaiseOnMouseMove(
   short button,
   short shift,
   int x,
   int y
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RaiseOnMouseMove(
   short button,
   short shift,
   float x,
   float y
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RaiseOnMouseMove(
   System.Object o1,
   System.Object o2,
   System.Object o3,
   System.Object o4
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RaiseOnMouseUp(
   short button,
   short shift,
   int x,
```

```
   int y
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RaiseOnMouseUp(
   short button,
   short shift,
   float x,
   float y
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RaiseOnMouseUp(
   System.Object o1,
   System.Object o2,
   System.Object o3,
   System.Object o4
);
protected void SetAboutBoxDelegate(
   System.Windows.Forms.AxHost+AboutBoxDelegate d
);
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
protected void SetVisibleCore(
   boolean value
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

AxHost.AboutBoxDelegate Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void AboutBoxDelegate();
```

AxHost.ActiveXInvokeKind Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ActiveXInvokeKind
```

Members

MethodInvoke

PropertyGet

PropertySet

AxHost.AxComponentEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AxComponentEditor
   extends System.Windows.Forms.Design.WindowsFormsComponentEditor
```

Public Constructors

```
public AxComponentEditor();
```

Public Methods

```
public boolean EditComponent(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object obj,
   System.Windows.Forms.IWin32Window parent
);
```

AxHost.ClsidAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, AllowMultiple = false, Inherit
ed = false) */
public class ClsidAttribute
   extends System.Attribute
```

Public Constructors

```
public ClsidAttribute(
   System.String clsid
);
```

Public Properties

```
public System.String get_Value();
```

AxHost.ConnectionPointCookie Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = false) */
public class ConnectionPointCookie
   extends System.Object
```

Public Constructors

```
public ConnectionPointCookie(
   System.Object source,
   System.Object sink,
   System.Type eventInterface
);
```

Public Methods

```
public void Disconnect();
```

Protected Methods

```
protected void Finalize();
```

AxHost.InvalidActiveXStateException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InvalidActiveXStateException
   extends System.Exception
```

Public Constructors

```
public InvalidActiveXStateException(
   System.String name,
   System.Windows.Forms.AxHost+ActiveXInvokeKind kind
);
```

Public Methods

```
public System.String ToString();
```

AxHost.State Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Compo
nentModel.TypeConverter, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77a5c56
1934e089") */
public class State
   extends System.Object
   implements System.Runtime.Serialization.ISerializable
```

Public Constructors

```
public State(
   System.IO.Stream ms,
   int storageType,
   boolean manualUpdate,
   System.String licKey
);
```

AxHost.StateConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class StateConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public StateConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

AxHost.TypeLibraryTimeStampAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Assembly, AllowMultiple = false, Inhe
rited = false) */
public class TypeLibraryTimeStampAttribute
   extends System.Attribute
```

Public Constructors

```
public TypeLibraryTimeStampAttribute(
   System.String timestamp
);
```

Public Properties

```
public System.DateTime get_Value();
```

BaseCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BaseCollection
   extends System.MarshalByRefObject
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public BaseCollection();
```

Public Properties

```
public int get_Count();
public boolean get_IsReadOnly();
public final boolean get_IsSynchronized();
public final System.Object get_SyncRoot();
```

Public Methods

```
public final void CopyTo(
   System.Array ar,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
```

Protected Properties

```
protected System.Collections.ArrayList get_List();
```

Binding Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Windo
ws.Forms.ListBindingConverter, System.Windows.Forms, Version=1.0.3300.0, Culture=neutral, Pub
licKeyToken=b77a5c561934e089") */
public class Binding
   extends System.Object
```

Public Constructors

```
public Binding(
   System.String propertyName,
   System.Object dataSource,
   System.String dataMember
);
```

Public Properties

```
public System.Windows.Forms.BindingManagerBase get_BindingManagerBase();
public System.Windows.Forms.BindingMemberInfo get_BindingMemberInfo();
public System.Windows.Forms.Control get_Control();
public System.Object get_DataSource();
public boolean get_IsBinding();
public System.String get_PropertyName();
```

Public Events

```
public void add_Format(
   System.Windows.Forms.ConvertEventHandler value
);
public void remove_Format(
   System.Windows.Forms.ConvertEventHandler value
);
public void add_Parse(
   System.Windows.Forms.ConvertEventHandler value
);
public void remove_Parse(
   System.Windows.Forms.ConvertEventHandler value
);
```

Protected Methods

```
protected void OnFormat(
   System.Windows.Forms.ConvertEventArgs cevent
);
protected void OnParse(
   System.Windows.Forms.ConvertEventArgs cevent
);
```

BindingContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "CollectionChanged") */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class BindingContext
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public BindingContext();
```

Public Properties

```
public boolean get_IsReadOnly();
public System.Windows.Forms.BindingManagerBase get_Item(
   System.Object dataSource
);
public System.Windows.Forms.BindingManagerBase get_Item(
   System.Object dataSource,
   System.String dataMember
);
```

Public Methods

```
public boolean Contains(
   System.Object dataSource
);
public boolean Contains(
   System.Object dataSource,
   System.String dataMember
);
```

Public Events

```
public void add_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
public void remove_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
```

Protected Methods

```
protected void Add(
   System.Object dataSource,
   System.Windows.Forms.BindingManagerBase listManager
);
protected void AddCore(
   System.Object dataSource,
   System.Windows.Forms.BindingManagerBase listManager
```

```
);
protected void Clear();
protected void ClearCore();
protected void OnCollectionChanged(
   System.ComponentModel.CollectionChangeEventArgs ccevent
);
protected void Remove(
   System.Object dataSource
);
protected void RemoveCore(
   System.Object dataSource
);
```

BindingManagerBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class BindingManagerBase
   extends System.Object
```

Public Constructors

```
public BindingManagerBase();
```

Public Properties

```
public System.Windows.Forms.BindingsCollection get_Bindings();
public abstract int get_Count();
public abstract System.Object get_Current();
public abstract int get_Position();
public abstract void set_Position(
   int value
);
```

Public Methods

```
public abstract void AddNew();
public abstract void CancelCurrentEdit();
public abstract void EndCurrentEdit();
public abstract System.ComponentModel.PropertyDescriptorCollection GetItemProperties();
public abstract void RemoveAt(
   int index
);
public abstract void ResumeBinding();
public abstract void SuspendBinding();
```

Public Events

```
public void add_CurrentChanged(
   System.EventHandler value
);
public void remove_CurrentChanged(
   System.EventHandler value
);
public void add_PositionChanged(
   System.EventHandler value
);
public void remove_PositionChanged(
   System.EventHandler value
);
```

Protected Methods

```
protected System.ComponentModel.PropertyDescriptorCollection GetItemProperties(
   System.Collections.ArrayList dataSources,
   System.Collections.ArrayList listAccessors
```

```
);
protected System.ComponentModel.PropertyDescriptorCollection GetItemProperties(
   System.Type listType,
   int offset,
   System.Collections.ArrayList dataSources,
   System.Collections.ArrayList listAccessors
);
protected abstract System.String GetListName(
   System.Collections.ArrayList listAccessors
);
protected abstract void OnCurrentChanged(
   System.EventArgs e
);
protected void PullData();
protected void PushData();
protected abstract void UpdateIsBinding();
```

Protected Fields

```
protected System.EventHandler onCurrentChangedHandler;
protected System.EventHandler onPositionChangedHandler;
```

BindingMemberInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class BindingMemberInfo
   extends System.ValueType
```

Public Constructors

```
public BindingMemberInfo(
   System.String dataMember
);
```

Public Properties

```
public System.String get_BindingField();
public System.String get_BindingMember();
public System.String get_BindingPath();
```

Public Methods

```
public boolean Equals(
   System.Object otherObject
);
public int GetHashCode();
```

BindingsCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "CollectionChanged") */
public class BindingsCollection
   extends System.Windows.Forms.BaseCollection
```

Public Properties

```
public int get_Count();
public System.Windows.Forms.Binding get_Item(
   int index
);
```

Public Events

```
public void add_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
public void remove_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
```

Protected Properties

```
protected System.Collections.ArrayList get_List();
```

Protected Methods

```
protected void Add(
   System.Windows.Forms.Binding binding
);
protected void AddCore(
   System.Windows.Forms.Binding dataBinding
);
protected void Clear();
protected void ClearCore();
protected void OnCollectionChanged(
   System.ComponentModel.CollectionChangeEventArgs ccevent
);
protected void Remove(
   System.Windows.Forms.Binding binding
);
protected void RemoveAt(
   int index
);
protected void RemoveCore(
   System.Windows.Forms.Binding dataBinding
);
protected boolean ShouldSerializeMyAll();
```

BootMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum BootMode
```

Members

FailSafe

FailSafeWithNetwork

Normal

Border3DSide Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.FlagsAttribute() */
public enum Border3DSide
```

Members

All

Bottom

Left

Middle

Right

Top

Border3DStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum Border3DStyle
```

Members

Adjust

Bump

Etched

Flat

Raised

RaisedInner

RaisedOuter

Sunken

SunkenInner

SunkenOuter

BorderStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum BorderStyle
```

Members

Fixed3D

FixedSingle

None

BoundsSpecified Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum BoundsSpecified
```

Members

All

Height

Location

None

Size

Width

X

Y

Button Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Button
   extends System.Windows.Forms.ButtonBase
   implements System.Windows.Forms.IButtonControl
```

Public Constructors

```
public Button();
```

Public Properties

```
public System.Windows.Forms.DialogResult get_DialogResult();
public void set_DialogResult(
   System.Windows.Forms.DialogResult value
);
```

Public Methods

```
public void NotifyDefault(
   boolean value
);
public final void PerformClick();
public System.String ToString();
```

Public Events

```
public void add_DoubleClick(
   System.EventHandler value
);
public void remove_DoubleClick(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
```

Protected Methods

```
protected void OnClick(
   System.EventArgs e
);
protected void OnMouseUp(
   System.Windows.Forms.MouseEventArgs mevent
);
protected boolean ProcessMnemonic(
   char charCode
);
```

```
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

ButtonBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class ButtonBase
   extends System.Windows.Forms.Control
```

Public Properties

```
public System.Windows.Forms.FlatStyle get_FlatStyle();
public void set_FlatStyle(
   System.Windows.Forms.FlatStyle value
);
public System.Drawing.Image get_Image();
public void set_Image(
   System.Drawing.Image value
);
public System.Drawing.ContentAlignment get_ImageAlign();
public void set_ImageAlign(
   System.Drawing.ContentAlignment value
);
public int get_ImageIndex();
public void set_ImageIndex(
   int value
);
public System.Windows.Forms.ImageList get_ImageList();
public void set_ImageList(
   System.Windows.Forms.ImageList value
);
public System.Windows.Forms.ImeMode get_ImeMode();
public void set_ImeMode(
   System.Windows.Forms.ImeMode value
);
public System.Drawing.ContentAlignment get_TextAlign();
public void set_TextAlign(
   System.Drawing.ContentAlignment value
);
```

Protected Constructors

```
protected ButtonBase();
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Windows.Forms.ImeMode get_DefaultImeMode();
protected System.Drawing.Size get_DefaultSize();
protected boolean get_IsDefault();
protected void set_IsDefault(
   boolean value
);
```

Protected Methods

```
protected System.Windows.Forms.AccessibleObject CreateAccessibilityInstance();
```

```
protected void Dispose(
   boolean disposing
);
protected void OnEnabledChanged(
   System.EventArgs e
);
protected void OnGotFocus(
   System.EventArgs e
);
protected void OnKeyDown(
   System.Windows.Forms.KeyEventArgs kevent
);
protected void OnKeyUp(
   System.Windows.Forms.KeyEventArgs kevent
);
protected void OnLostFocus(
   System.EventArgs e
);
protected void OnMouseDown(
   System.Windows.Forms.MouseEventArgs mevent
);
protected void OnMouseEnter(
   System.EventArgs eventargs
);
protected void OnMouseLeave(
   System.EventArgs eventargs
);
protected void OnMouseMove(
   System.Windows.Forms.MouseEventArgs mevent
);
protected void OnMouseUp(
   System.Windows.Forms.MouseEventArgs mevent
);
protected void OnPaint(
   System.Windows.Forms.PaintEventArgs pevent
);
protected void OnParentChanged(
   System.EventArgs e
);
protected void OnTextChanged(
   System.EventArgs e
);
protected void OnVisibleChanged(
   System.EventArgs e
);
protected void ResetFlagsandPaint();
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

ButtonBase.ButtonBaseAccessibleObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class ButtonBaseAccessibleObject
   extends System.Windows.Forms.Control+ControlAccessibleObject
```

Public Constructors

```
public ButtonBaseAccessibleObject(
   System.Windows.Forms.Control owner
);
```

Public Methods

```
public void DoDefaultAction();
```

ButtonBorderStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ButtonBorderStyle
```

Members

Dashed

Dotted

Inset

None

Outset

Solid

ButtonState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum ButtonState
```

Members

All

Checked

Flat

Inactive

Normal

Pushed

CaptionButton Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CaptionButton
```

Members

Close

Help

Maximize

Minimize

Restore

CharacterCasing Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CharacterCasing
```

Members

Lower

Normal

Upper

CheckBox Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "CheckedChanged") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Checked") */
public class CheckBox
   extends System.Windows.Forms.ButtonBase
```

Public Constructors

```
public CheckBox();
```

Public Properties

```
public System.Windows.Forms.Appearance get_Appearance();
public void set_Appearance(
   System.Windows.Forms.Appearance value
);
public boolean get_AutoCheck();
public void set_AutoCheck(
   boolean value
);
public System.Drawing.ContentAlignment get_CheckAlign();
public void set_CheckAlign(
   System.Drawing.ContentAlignment value
);
public boolean get_Checked();
public void set_Checked(
   boolean value
);
public System.Windows.Forms.CheckState get_CheckState();
public void set_CheckState(
   System.Windows.Forms.CheckState value
);
public System.Drawing.ContentAlignment get_TextAlign();
public void set_TextAlign(
   System.Drawing.ContentAlignment value
);
public boolean get_ThreeState();
public void set_ThreeState(
   boolean value
);
```

Public Methods

```
public System.String ToString();
```

Public Events

```
public void add_AppearanceChanged(
   System.EventHandler value
);
public void remove_AppearanceChanged(
   System.EventHandler value
```

```
);
public void add_CheckedChanged(
   System.EventHandler value
);
public void remove_CheckedChanged(
   System.EventHandler value
);
public void add_CheckStateChanged(
   System.EventHandler value
);
public void remove_CheckStateChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected System.Windows.Forms.AccessibleObject CreateAccessibilityInstance();
protected void OnAppearanceChanged(
   System.EventArgs e
);
protected void OnCheckedChanged(
   System.EventArgs e
);
protected void OnCheckStateChanged(
   System.EventArgs e
);
protected void OnClick(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnMouseUp(
   System.Windows.Forms.MouseEventArgs mevent
);
protected boolean ProcessMnemonic(
   char charCode
);
```

CheckBox.CheckBoxAccessibleObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class CheckBoxAccessibleObject
   extends System.Windows.Forms.ButtonBase+ButtonBaseAccessibleObject
```

Public Constructors

```
public CheckBoxAccessibleObject(
   System.Windows.Forms.Control owner
);
```

Public Properties

```
public System.String get_DefaultAction();
public System.Windows.Forms.AccessibleRole get_Role();
public System.Windows.Forms.AccessibleStates get_State();
```

CheckedListBox Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CheckedListBox
   extends System.Windows.Forms.ListBox
```

Public Constructors

```
public CheckedListBox();
```

Public Properties

```
public System.Windows.Forms.CheckedListBox+CheckedIndexCollection get_CheckedIndices();
public System.Windows.Forms.CheckedListBox+CheckedItemCollection get_CheckedItems();
public boolean get_CheckOnClick();
public void set_CheckOnClick(
   boolean value
);
public System.Windows.Forms.DrawMode get_DrawMode();
public void set_DrawMode(
   System.Windows.Forms.DrawMode value
);
public int get_ItemHeight();
public void set_ItemHeight(
   int value
);
public System.Windows.Forms.CheckedListBox+ObjectCollection get_Items();
public System.Windows.Forms.SelectionMode get_SelectionMode();
public void set_SelectionMode(
   System.Windows.Forms.SelectionMode value
);
public boolean get_ThreeDCheckBoxes();
public void set_ThreeDCheckBoxes(
   boolean value
);
```

Public Methods

```
public boolean GetItemChecked(
   int index
);
public System.Windows.Forms.CheckState GetItemCheckState(
   int index
);
public void SetItemChecked(
   int index,
   boolean value
);
public void SetItemCheckState(
   int index,
   System.Windows.Forms.CheckState value
);
```

Public Events

```
public void add_Click(
   System.EventHandler value
);
public void remove_Click(
   System.EventHandler value
);
public void add_DrawItem(
   System.Windows.Forms.DrawItemEventHandler value
);
public void remove_DrawItem(
   System.Windows.Forms.DrawItemEventHandler value
);
public void add_ItemCheck(
   System.Windows.Forms.ItemCheckEventHandler value
);
public void remove_ItemCheck(
   System.Windows.Forms.ItemCheckEventHandler value
);
public void add_MeasureItem(
   System.Windows.Forms.MeasureItemEventHandler value
);
public void remove_MeasureItem(
   System.Windows.Forms.MeasureItemEventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
```

Protected Methods

```
protected System.Windows.Forms.AccessibleObject CreateAccessibilityInstance();
protected System.Windows.Forms.ListBox+ObjectCollection CreateItemCollection();
protected void OnBackColorChanged(
   System.EventArgs e
);
protected void OnClick(
   System.EventArgs e
);
protected void OnDrawItem(
   System.Windows.Forms.DrawItemEventArgs e
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnItemCheck(
   System.Windows.Forms.ItemCheckEventArgs ice
);
protected void OnKeyPress(
   System.Windows.Forms.KeyPressEventArgs e
);
protected void OnMeasureItem(
   System.Windows.Forms.MeasureItemEventArgs e
);
protected void OnSelectedIndexChanged(
   System.EventArgs e
);
protected void WmReflectCommand(
   System.Windows.Forms.Message& m
);
protected void WndProc(
```

```
    System.Windows.Forms.Message& m
);
```

CheckedListBox.CheckedIndexCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CheckedIndexCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public int get_Item(
   int index
);
```

Public Methods

```
public boolean Contains(
   int index
);
public final void CopyTo(
   System.Array dest,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   int index
);
```

CheckedListBox.CheckedItemCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CheckedItemCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public final System.Object get_Item(
   int index
);
public final void set_Item(
   int index,
   System.Object value
);
```

Public Methods

```
public final boolean Contains(
   System.Object item
);
public final void CopyTo(
   System.Array dest,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public final int IndexOf(
   System.Object item
);
```

CheckedListBox.ObjectCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ObjectCollection
   extends System.Windows.Forms.ListBox+ObjectCollection
```

Public Constructors

```
public ObjectCollection(
   System.Windows.Forms.CheckedListBox owner
);
```

Public Methods

```
public int Add(
   System.Object item,
   boolean isChecked
);
public int Add(
   System.Object item,
   System.Windows.Forms.CheckState check
);
```

CheckState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum CheckState
```

Members

Checked

Indeterminate

Unchecked

Clipboard Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Clipboard
   extends System.Object
```

Public Methods

```
public static System.Windows.Forms.IDataObject GetDataObject();
public static void SetDataObject(
   System.Object data
);
public static void SetDataObject(
   System.Object data,
   boolean copy
);
```

ColorDepth Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ColorDepth
```

Members

Depth16Bit

Depth24Bit

Depth32Bit

Depth4Bit

Depth8Bit

ColorDialog Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Color") */
public class ColorDialog
   extends System.Windows.Forms.CommonDialog
```

Public Constructors

```
public ColorDialog();
```

Public Properties

```
public boolean get_AllowFullOpen();
public void set_AllowFullOpen(
   boolean value
);
public boolean get_AnyColor();
public void set_AnyColor(
   boolean value
);
public System.Drawing.Color get_Color();
public void set_Color(
   System.Drawing.Color value
);
public int[] get_CustomColors();
public void set_CustomColors(
   int[] value
);
public boolean get_FullOpen();
public void set_FullOpen(
   boolean value
);
public boolean get_ShowHelp();
public void set_ShowHelp(
   boolean value
);
public boolean get_SolidColorOnly();
public void set_SolidColorOnly(
   boolean value
);
```

Public Methods

```
public void Reset();
public System.String ToString();
```

Protected Properties

```
protected System.IntPtr get_Instance();
protected int get_Options();
```

Protected Methods

```
protected boolean RunDialog(
   System.IntPtr hwndOwner
);
```

ColumnClickEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ColumnClickEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ColumnClickEventArgs(
   int column
);
```

Public Properties

```
public int get_Column();
```

ColumnClickEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ColumnClickEventHandler(
   System.Object sender,
   System.Windows.Forms.ColumnClickEventArgs e
);
```

ColumnHeader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
public class ColumnHeader
   extends System.ComponentModel.Component
   implements System.ICloneable
```

Public Constructors

```
public ColumnHeader();
```

Public Properties

```
public int get_Index();
public System.Windows.Forms.ListView get_ListView();
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.Windows.Forms.HorizontalAlignment get_TextAlign();
public void set_TextAlign(
   System.Windows.Forms.HorizontalAlignment value
);
public int get_Width();
public void set_Width(
   int value
);
```

Public Methods

```
public final System.Object Clone();
public System.String ToString();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

ColumnHeaderStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ColumnHeaderStyle
```

Members

Clickable

Nonclickable

None

ComboBox Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.ComboBoxDesigner,
System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Items") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "SelectedIndexChanged") */
public class ComboBox
   extends System.Windows.Forms.ListControl
```

Public Constructors

```
public ComboBox();
```

Public Properties

```
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.DrawMode get_DrawMode();
public void set_DrawMode(
   System.Windows.Forms.DrawMode value
);
public System.Windows.Forms.ComboBoxStyle get_DropDownStyle();
public void set_DropDownStyle(
   System.Windows.Forms.ComboBoxStyle value
);
public int get_DropDownWidth();
public void set_DropDownWidth(
   int value
);
public boolean get_DroppedDown();
public void set_DroppedDown(
   boolean value
);
public boolean get_Focused();
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public boolean get_IntegralHeight();
public void set_IntegralHeight(
   boolean value
);
public int get_ItemHeight();
public void set_ItemHeight(
   int value
);
public System.Windows.Forms.ComboBox+ObjectCollection get_Items();
public int get_MaxDropDownItems();
public void set_MaxDropDownItems(
   int value
```

```
);
public int get_MaxLength();
public void set_MaxLength(
   int value
);
public int get_PreferredHeight();
public int get_SelectedIndex();
public void set_SelectedIndex(
   int value
);
public System.Object get_SelectedItem();
public void set_SelectedItem(
   System.Object value
);
public System.String get_SelectedText();
public void set_SelectedText(
   System.String value
);
public int get_SelectionLength();
public void set_SelectionLength(
   int value
);
public int get_SelectionStart();
public void set_SelectionStart(
   int value
);
public boolean get_Sorted();
public void set_Sorted(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Methods

```
public void BeginUpdate();
public void EndUpdate();
public int FindString(
   System.String s
);
public int FindString(
   System.String s,
   int startIndex
);
public int FindStringExact(
   System.String s
);
public int FindStringExact(
   System.String s,
   int startIndex
);
public int GetItemHeight(
   int index
);
public void Select(
   int start,
   int length
);
public void SelectAll();
public System.String ToString();
```

Public Events

```
public void add_DrawItem(
   System.Windows.Forms.DrawItemEventHandler value
);
public void remove_DrawItem(
   System.Windows.Forms.DrawItemEventHandler value
);
public void add_DropDown(
   System.EventHandler value
);
public void remove_DropDown(
   System.EventHandler value
);
public void add_DropDownStyleChanged(
   System.EventHandler value
);
public void remove_DropDownStyleChanged(
   System.EventHandler value
);
public void add_MeasureItem(
   System.Windows.Forms.MeasureItemEventHandler value
);
public void remove_MeasureItem(
   System.Windows.Forms.MeasureItemEventHandler value
);
public void add_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void add_SelectedIndexChanged(
   System.EventHandler value
);
public void remove_SelectedIndexChanged(
   System.EventHandler value
);
public void add_SelectionChangeCommitted(
   System.EventHandler value
);
public void remove_SelectionChangeCommitted(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void AddItemsCore(
   System.Object[] value
);
protected void Dispose(
   boolean disposing
);
protected boolean IsInputKey(
   System.Windows.Forms.Keys keyData
);
protected void OnBackColorChanged(
   System.EventArgs e
);
protected void OnDataSourceChanged(
   System.EventArgs e
```

```
);
protected void OnDisplayMemberChanged(
   System.EventArgs e
);
protected void OnDrawItem(
   System.Windows.Forms.DrawItemEventArgs e
);
protected void OnDropDown(
   System.EventArgs e
);
protected void OnDropDownStyleChanged(
   System.EventArgs e
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnForeColorChanged(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnHandleDestroyed(
   System.EventArgs e
);
protected void OnKeyPress(
   System.Windows.Forms.KeyPressEventArgs e
);
protected void OnMeasureItem(
   System.Windows.Forms.MeasureItemEventArgs e
);
protected void OnParentBackColorChanged(
   System.EventArgs e
);
protected void OnResize(
   System.EventArgs e
);
protected void OnSelectedIndexChanged(
   System.EventArgs e
);
protected void OnSelectedItemChanged(
   System.EventArgs e
);
protected void OnSelectionChangeCommitted(
   System.EventArgs e
);
protected void RefreshItem(
   int index
);
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
protected void SetItemCore(
   int index,
   System.Object value
);
protected void SetItemsCore(
   System.Collections.IList value
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

ComboBox.ChildAccessibleObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class ChildAccessibleObject
   extends System.Windows.Forms.AccessibleObject
```

Public Constructors

```
public ChildAccessibleObject(
   System.Windows.Forms.ComboBox owner,
   System.IntPtr handle
);
```

Public Properties

```
public System.String get_Name();
```

ComboBox.ObjectCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.ListBindableAttribute(ListBindable = false) */
public class ObjectCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public ObjectCollection(
   System.Windows.Forms.ComboBox owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public System.Object get_Item(
   int index
);
public void set_Item(
   int index,
   System.Object value
);
```

Public Methods

```
public int Add(
   System.Object item
);
public void AddRange(
   System.Object[] items
);
public final void Clear();
public final boolean Contains(
   System.Object value
);
public void CopyTo(
   System.Object[] dest,
   int arrayIndex
);
public final System.Collections.IEnumerator GetEnumerator();
public final int IndexOf(
   System.Object value
);
public final void Insert(
   int index,
   System.Object item
);
public final void Remove(
   System.Object value
);
public final void RemoveAt(
```

```
    int index
);
```

ComboBoxStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ComboBoxStyle
```

Members

DropDown

DropDownList

Simple

CommonDialog Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemFilterAttribute(FilterString = "System.Window
s.Forms", FilterType = Allow) */
public abstract class CommonDialog
   extends System.ComponentModel.Component
```

Public Constructors

```
public CommonDialog();
```

Public Methods

```
public abstract void Reset();
public System.Windows.Forms.DialogResult ShowDialog();
public System.Windows.Forms.DialogResult ShowDialog(
   System.Windows.Forms.IWin32Window owner
);
```

Public Events

```
public void add_HelpRequest(
   System.EventHandler value
);
public void remove_HelpRequest(
   System.EventHandler value
);
```

Protected Methods

```
protected System.IntPtr HookProc(
   System.IntPtr hWnd,
   int msg,
   System.IntPtr wparam,
   System.IntPtr lparam
);
protected void OnHelpRequest(
   System.EventArgs e
);
protected System.IntPtr OwnerWndProc(
   System.IntPtr hWnd,
   int msg,
   System.IntPtr wparam,
   System.IntPtr lparam
);
protected abstract boolean RunDialog(
   System.IntPtr hwndOwner
);
```

ContainerControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ContainerControl
   extends System.Windows.Forms.ScrollableControl
   implements System.Windows.Forms.IContainerControl
```

Public Constructors

```
public ContainerControl();
```

Public Properties

```
public final System.Windows.Forms.Control get_ActiveControl();
public final void set_ActiveControl(
   System.Windows.Forms.Control value
);
public System.Windows.Forms.BindingContext get_BindingContext();
public void set_BindingContext(
   System.Windows.Forms.BindingContext value
);
public System.Windows.Forms.Form get_ParentForm();
```

Public Methods

```
public boolean Validate();
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
```

Protected Methods

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void AdjustFormScrollbars(
   boolean displayScrollbars
);
protected void Dispose(
   boolean disposing
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnControlRemoved(
   System.Windows.Forms.ControlEventArgs e
);
protected void OnCreateControl();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected boolean ProcessDialogChar(
   char charCode
);
protected boolean ProcessDialogKey(
   System.Windows.Forms.Keys keyData
);
```

```
protected boolean ProcessMnemonic(
   char charCode
);
protected boolean ProcessTabKey(
   boolean forward
);
protected void Select(
   boolean directed,
   boolean forward
);
protected void UpdateDefaultButton();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

ContentsResizedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ContentsResizedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ContentsResizedEventArgs(
   System.Drawing.Rectangle newRectangle
);
```

Public Properties

```
public System.Drawing.Rectangle get_NewRectangle();
```

ContentsResizedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ContentsResizedEventHandler(
   System.Object sender,
   System.Windows.Forms.ContentsResizedEventArgs e
);
```

ContextMenu Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Popup") */
public class ContextMenu
   extends System.Windows.Forms.Menu
```

Public Constructors

```
public ContextMenu();
public ContextMenu(
   System.Windows.Forms.MenuItem[] menuItems
);
```

Public Properties

```
public System.Windows.Forms.RightToLeft get_RightToLeft();
public void set_RightToLeft(
   System.Windows.Forms.RightToLeft value
);
public System.Windows.Forms.Control get_SourceControl();
```

Public Methods

```
public void Show(
   System.Windows.Forms.Control control,
   System.Drawing.Point pos
);
```

Public Events

```
public void add_Popup(
   System.EventHandler value
);
public void remove_Popup(
   System.EventHandler value
);
```

Protected Methods

```
protected void OnPopup(
   System.EventArgs e
);
```

Control Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.ControlDesigner, S
ystem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.ToolboxItemFilterAttribute(FilterString = "System.Window
s.Forms", FilterType = Allow) */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Click") */
/** @attribute System.ComponentModel.Design.Serialization.DesignerSerializerAttribute(Seriali
zerTypeName = "System.Windows.Forms.Design.ControlCodeDomSerializer, System.Design, Version=1
.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a", SerializerBaseTypeName = "Syste
m.ComponentModel.Design.Serialization.CodeDomSerializer, System.Design, Version=1.0.3300.0, C
ulture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class Control
   extends System.ComponentModel.Component
   implements System.ComponentModel.ISynchronizeInvoke, System.Windows.Forms.IWin32Window
```

Public Constructors

```
public Control();
public Control(
   System.String text
);
public Control(
   System.Windows.Forms.Control parent,
   System.String text
);
public Control(
   System.String text,
   int left,
   int top,
   int width,
   int height
);
public Control(
   System.Windows.Forms.Control parent,
   System.String text,
   int left,
   int top,
   int width,
   int height
);
```

Public Properties

```
public System.Windows.Forms.AccessibleObject get_AccessibilityObject();
public System.String get_AccessibleDefaultActionDescription();
public void set_AccessibleDefaultActionDescription(
   System.String value
);
public System.String get_AccessibleDescription();
public void set_AccessibleDescription(
   System.String value
);
public System.String get_AccessibleName();
public void set_AccessibleName(
```

```
   System.String value
);
public System.Windows.Forms.AccessibleRole get_AccessibleRole();
public void set_AccessibleRole(
   System.Windows.Forms.AccessibleRole value
);
public boolean get_AllowDrop();
public void set_AllowDrop(
   boolean value
);
public System.Windows.Forms.AnchorStyles get_Anchor();
public void set_Anchor(
   System.Windows.Forms.AnchorStyles value
);
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.BindingContext get_BindingContext();
public void set_BindingContext(
   System.Windows.Forms.BindingContext value
);
public int get_Bottom();
public System.Drawing.Rectangle get_Bounds();
public void set_Bounds(
   System.Drawing.Rectangle value
);
public boolean get_CanFocus();
public boolean get_CanSelect();
public boolean get_Capture();
public void set_Capture(
   boolean value
);
public boolean get_CausesValidation();
public void set_CausesValidation(
   boolean value
);
public System.Drawing.Rectangle get_ClientRectangle();
public System.Drawing.Size get_ClientSize();
public void set_ClientSize(
   System.Drawing.Size value
);
public System.String get_CompanyName();
public boolean get_ContainsFocus();
public System.Windows.Forms.ContextMenu get_ContextMenu();
public void set_ContextMenu(
   System.Windows.Forms.ContextMenu value
);
public System.Windows.Forms.Control+ControlCollection get_Controls();
public boolean get_Created();
public System.Windows.Forms.Cursor get_Cursor();
public void set_Cursor(
   System.Windows.Forms.Cursor value
);
public System.Windows.Forms.ControlBindingsCollection get_DataBindings();
public static System.Drawing.Color get_DefaultBackColor();
public static System.Drawing.Font get_DefaultFont();
public static System.Drawing.Color get_DefaultForeColor();
public System.Drawing.Rectangle get_DisplayRectangle();
public boolean get_Disposing();
public System.Windows.Forms.DockStyle get_Dock();
public void set_Dock(
   System.Windows.Forms.DockStyle value
);
```

```
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public boolean get_Focused();
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public final System.IntPtr get_Handle();
public boolean get_HasChildren();
public int get_Height();
public void set_Height(
   int value
);
public System.Windows.Forms.ImeMode get_ImeMode();
public void set_ImeMode(
   System.Windows.Forms.ImeMode value
);
public final boolean get_InvokeRequired();
public boolean get_IsAccessible();
public void set_IsAccessible(
   boolean value
);
public boolean get_IsDisposed();
public boolean get_IsHandleCreated();
public int get_Left();
public void set_Left(
   int value
);
public System.Drawing.Point get_Location();
public void set_Location(
   System.Drawing.Point value
);
public static System.Windows.Forms.Keys get_ModifierKeys();
public static System.Windows.Forms.MouseButtons get_MouseButtons();
public static System.Drawing.Point get_MousePosition();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Windows.Forms.Control get_Parent();
public void set_Parent(
   System.Windows.Forms.Control value
);
public System.String get_ProductName();
public System.String get_ProductVersion();
public boolean get_RecreatingHandle();
public System.Drawing.Region get_Region();
public void set_Region(
   System.Drawing.Region value
);
public int get_Right();
public System.Windows.Forms.RightToLeft get_RightToLeft();
public void set_RightToLeft(
   System.Windows.Forms.RightToLeft value
);
public System.ComponentModel.ISite get_Site();
public void set_Site(
   System.ComponentModel.ISite value
);
public System.Drawing.Size get_Size();
public void set_Size(
   System.Drawing.Size value
```

```
);
public int get_TabIndex();
public void set_TabIndex(
   int value
);
public boolean get_TabStop();
public void set_TabStop(
   boolean value
);
public System.Object get_Tag();
public void set_Tag(
   System.Object value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public int get_Top();
public void set_Top(
   int value
);
public System.Windows.Forms.Control get_TopLevelControl();
public boolean get_Visible();
public void set_Visible(
   boolean value
);
public int get_Width();
public void set_Width(
   int value
);
public System.Windows.Forms.IWindowTarget get_WindowTarget();
public void set_WindowTarget(
   System.Windows.Forms.IWindowTarget value
);
```

Public Methods

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public System.IAsyncResult BeginInvoke(
   System.Delegate method
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public final System.IAsyncResult BeginInvoke(
   System.Delegate method,
   System.Object[] args
);
public void BringToFront();
public boolean Contains(
   System.Windows.Forms.Control ctl
);
public void CreateControl();
public System.Drawing.Graphics CreateGraphics();
public System.Windows.Forms.DragDropEffects DoDragDrop(
   System.Object data,
   System.Windows.Forms.DragDropEffects allowedEffects
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public final System.Object EndInvoke(
   System.IAsyncResult asyncResult
);
public System.Windows.Forms.Form FindForm();
public boolean Focus();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public static System.Windows.Forms.Control FromChildHandle(
   System.IntPtr handle
);
```

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public static System.Windows.Forms.Control FromHandle(
   System.IntPtr handle
);
public System.Windows.Forms.Control GetChildAtPoint(
   System.Drawing.Point pt
);
public System.Windows.Forms.IContainerControl GetContainerControl();
public System.Windows.Forms.Control GetNextControl(
   System.Windows.Forms.Control ctl,
   boolean forward
);
public void Hide();
public void Invalidate();
public void Invalidate(
   boolean invalidateChildren
);
public void Invalidate(
   System.Drawing.Rectangle rc
);
public void Invalidate(
   System.Drawing.Region region
);
public void Invalidate(
   System.Drawing.Rectangle rc,
   boolean invalidateChildren
);
public void Invalidate(
   System.Drawing.Region region,
   boolean invalidateChildren
);
public System.Object Invoke(
   System.Delegate method
);
public final System.Object Invoke(
   System.Delegate method,
   System.Object[] args
);
public static boolean IsMnemonic(
   char charCode,
   System.String text
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public void PerformLayout();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public void PerformLayout(
   System.Windows.Forms.Control affectedControl,
   System.String affectedProperty
);
public System.Drawing.Point PointToClient(
   System.Drawing.Point p
);
public System.Drawing.Point PointToScreen(
   System.Drawing.Point p
);
public boolean PreProcessMessage(
   System.Windows.Forms.Message& msg
);
public System.Drawing.Rectangle RectangleToClient(
   System.Drawing.Rectangle r
);
public System.Drawing.Rectangle RectangleToScreen(
   System.Drawing.Rectangle r
);
public void Refresh();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public void ResetBackColor();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
```

```
public void ResetBindings();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public void ResetCursor();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public void ResetFont();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public void ResetForeColor();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public void ResetImeMode();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public void ResetRightToLeft();
public void ResetText();
public void ResumeLayout();
public void ResumeLayout(
   boolean performLayout
);
public void Scale(
   float ratio
);
public void Scale(
   float dx,
   float dy
);
public void Select();
public boolean SelectNextControl(
   System.Windows.Forms.Control ctl,
   boolean forward,
   boolean tabStopOnly,
   boolean nested,
   boolean wrap
);
public void SendToBack();
public void SetBounds(
   int x,
   int y,
   int width,
   int height
);
public void SetBounds(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
public void Show();
public void SuspendLayout();
public void Update();
```

Public Events

```
public void add_BackColorChanged(
   System.EventHandler value
);
public void remove_BackColorChanged(
   System.EventHandler value
);
public void add_BackgroundImageChanged(
   System.EventHandler value
);
public void remove_BackgroundImageChanged(
   System.EventHandler value
);
public void add_BindingContextChanged(
   System.EventHandler value
);
```

```
public void remove_BindingContextChanged(
   System.EventHandler value
);
public void add_CausesValidationChanged(
   System.EventHandler value
);
public void remove_CausesValidationChanged(
   System.EventHandler value
);
public void add_ChangeUICues(
   System.Windows.Forms.UICuesEventHandler value
);
public void remove_ChangeUICues(
   System.Windows.Forms.UICuesEventHandler value
);
public void add_Click(
   System.EventHandler value
);
public void remove_Click(
   System.EventHandler value
);
public void add_ContextMenuChanged(
   System.EventHandler value
);
public void remove_ContextMenuChanged(
   System.EventHandler value
);
public void add_ControlAdded(
   System.Windows.Forms.ControlEventHandler value
);
public void remove_ControlAdded(
   System.Windows.Forms.ControlEventHandler value
);
public void add_ControlRemoved(
   System.Windows.Forms.ControlEventHandler value
);
public void remove_ControlRemoved(
   System.Windows.Forms.ControlEventHandler value
);
public void add_CursorChanged(
   System.EventHandler value
);
public void remove_CursorChanged(
   System.EventHandler value
);
public void add_DockChanged(
   System.EventHandler value
);
public void remove_DockChanged(
   System.EventHandler value
);
public void add_DoubleClick(
   System.EventHandler value
);
public void remove_DoubleClick(
   System.EventHandler value
);
public void add_DragDrop(
   System.Windows.Forms.DragEventHandler value
);
public void remove_DragDrop(
   System.Windows.Forms.DragEventHandler value
);
public void add_DragEnter(
   System.Windows.Forms.DragEventHandler value
);
public void remove_DragEnter(
   System.Windows.Forms.DragEventHandler value
```

```
);
public void add_DragLeave(
   System.EventHandler value
);
public void remove_DragLeave(
   System.EventHandler value
);
public void add_DragOver(
   System.Windows.Forms.DragEventHandler value
);
public void remove_DragOver(
   System.Windows.Forms.DragEventHandler value
);
public void add_EnabledChanged(
   System.EventHandler value
);
public void remove_EnabledChanged(
   System.EventHandler value
);
public void add_Enter(
   System.EventHandler value
);
public void remove_Enter(
   System.EventHandler value
);
public void add_FontChanged(
   System.EventHandler value
);
public void remove_FontChanged(
   System.EventHandler value
);
public void add_ForeColorChanged(
   System.EventHandler value
);
public void remove_ForeColorChanged(
   System.EventHandler value
);
public void add_GiveFeedback(
   System.Windows.Forms.GiveFeedbackEventHandler value
);
public void remove_GiveFeedback(
   System.Windows.Forms.GiveFeedbackEventHandler value
);
public void add_GotFocus(
   System.EventHandler value
);
public void remove_GotFocus(
   System.EventHandler value
);
public void add_HandleCreated(
   System.EventHandler value
);
public void remove_HandleCreated(
   System.EventHandler value
);
public void add_HandleDestroyed(
   System.EventHandler value
);
public void remove_HandleDestroyed(
   System.EventHandler value
);
public void add_HelpRequested(
   System.Windows.Forms.HelpEventHandler value
);
public void remove_HelpRequested(
   System.Windows.Forms.HelpEventHandler value
);
public void add_ImeModeChanged(
```

```
    System.EventHandler value
);
public void remove_ImeModeChanged(
    System.EventHandler value
);
public void add_Invalidated(
    System.Windows.Forms.InvalidateEventHandler value
);
public void remove_Invalidated(
    System.Windows.Forms.InvalidateEventHandler value
);
public void add_KeyDown(
    System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyDown(
    System.Windows.Forms.KeyEventHandler value
);
public void add_KeyPress(
    System.Windows.Forms.KeyPressEventHandler value
);
public void remove_KeyPress(
    System.Windows.Forms.KeyPressEventHandler value
);
public void add_KeyUp(
    System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyUp(
    System.Windows.Forms.KeyEventHandler value
);
public void add_Layout(
    System.Windows.Forms.LayoutEventHandler value
);
public void remove_Layout(
    System.Windows.Forms.LayoutEventHandler value
);
public void add_Leave(
    System.EventHandler value
);
public void remove_Leave(
    System.EventHandler value
);
public void add_LocationChanged(
    System.EventHandler value
);
public void remove_LocationChanged(
    System.EventHandler value
);
public void add_LostFocus(
    System.EventHandler value
);
public void remove_LostFocus(
    System.EventHandler value
);
public void add_MouseDown(
    System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseDown(
    System.Windows.Forms.MouseEventHandler value
);
public void add_MouseEnter(
    System.EventHandler value
);
public void remove_MouseEnter(
    System.EventHandler value
);
public void add_MouseHover(
    System.EventHandler value
);
```

```
public void remove_MouseHover(
   System.EventHandler value
);
public void add_MouseLeave(
   System.EventHandler value
);
public void remove_MouseLeave(
   System.EventHandler value
);
public void add_MouseMove(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseMove(
   System.Windows.Forms.MouseEventHandler value
);
public void add_MouseUp(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseUp(
   System.Windows.Forms.MouseEventHandler value
);
public void add_MouseWheel(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseWheel(
   System.Windows.Forms.MouseEventHandler value
);
public void add_Move(
   System.EventHandler value
);
public void remove_Move(
   System.EventHandler value
);
public void add_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void add_ParentChanged(
   System.EventHandler value
);
public void remove_ParentChanged(
   System.EventHandler value
);
public void add_QueryAccessibilityHelp(
   System.Windows.Forms.QueryAccessibilityHelpEventHandler value
);
public void remove_QueryAccessibilityHelp(
   System.Windows.Forms.QueryAccessibilityHelpEventHandler value
);
public void add_QueryContinueDrag(
   System.Windows.Forms.QueryContinueDragEventHandler value
);
public void remove_QueryContinueDrag(
   System.Windows.Forms.QueryContinueDragEventHandler value
);
public void add_Resize(
   System.EventHandler value
);
public void remove_Resize(
   System.EventHandler value
);
public void add_RightToLeftChanged(
   System.EventHandler value
);
public void remove_RightToLeftChanged(
   System.EventHandler value
```

```
);
public void add_SizeChanged(
   System.EventHandler value
);
public void remove_SizeChanged(
   System.EventHandler value
);
public void add_StyleChanged(
   System.EventHandler value
);
public void remove_StyleChanged(
   System.EventHandler value
);
public void add_SystemColorsChanged(
   System.EventHandler value
);
public void remove_SystemColorsChanged(
   System.EventHandler value
);
public void add_TabIndexChanged(
   System.EventHandler value
);
public void remove_TabIndexChanged(
   System.EventHandler value
);
public void add_TabStopChanged(
   System.EventHandler value
);
public void remove_TabStopChanged(
   System.EventHandler value
);
public void add_TextChanged(
   System.EventHandler value
);
public void remove_TextChanged(
   System.EventHandler value
);
public void add_Validated(
   System.EventHandler value
);
public void remove_Validated(
   System.EventHandler value
);
public void add_Validating(
   System.ComponentModel.CancelEventHandler value
);
public void remove_Validating(
   System.ComponentModel.CancelEventHandler value
);
public void add_VisibleChanged(
   System.EventHandler value
);
public void remove_VisibleChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Windows.Forms.ImeMode get_DefaultImeMode();
protected System.Drawing.Size get_DefaultSize();
protected int get_FontHeight();
protected void set_FontHeight(
   int value
);
protected boolean get_RenderRightToLeft();
```

```
protected boolean get_ResizeRedraw();
protected void set_ResizeRedraw(
   boolean value
);
protected boolean get_ShowFocusCues();
protected boolean get_ShowKeyboardCues();
```

Protected Methods

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void AccessibilityNotifyClients(
   System.Windows.Forms.AccessibleEvents accEvent,
   int childID
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected System.Windows.Forms.AccessibleObject CreateAccessibilityInstance();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected System.Windows.Forms.Control+ControlCollection CreateControlsInstance();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void CreateHandle();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void DefWndProc(
   System.Windows.Forms.Message& m
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void DestroyHandle();
protected void Dispose(
   boolean disposing
);
protected boolean GetStyle(
   System.Windows.Forms.ControlStyles flag
);
protected boolean GetTopLevel();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void InitLayout();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void InvokeGotFocus(
   System.Windows.Forms.Control toInvoke,
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void InvokeLostFocus(
   System.Windows.Forms.Control toInvoke,
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void InvokeOnClick(
   System.Windows.Forms.Control toInvoke,
   System.EventArgs e
);
protected void InvokePaint(
   System.Windows.Forms.Control c,
   System.Windows.Forms.PaintEventArgs e
);
protected void InvokePaintBackground(
   System.Windows.Forms.Control c,
   System.Windows.Forms.PaintEventArgs e
);
protected boolean IsInputChar(
   char charCode
);
protected boolean IsInputKey(
   System.Windows.Forms.Keys keyData
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void NotifyInvalidate(
```

```
   System.Drawing.Rectangle invalidatedArea
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnBackColorChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnBackgroundImageChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnBindingContextChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnCausesValidationChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnChangeUICues(
   System.Windows.Forms.UICuesEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnClick(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnContextMenuChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnControlAdded(
   System.Windows.Forms.ControlEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnControlRemoved(
   System.Windows.Forms.ControlEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnCreateControl();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnCursorChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnDockChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnDoubleClick(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnDragDrop(
   System.Windows.Forms.DragEventArgs drgevent
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnDragEnter(
   System.Windows.Forms.DragEventArgs drgevent
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnDragLeave(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnDragOver(
   System.Windows.Forms.DragEventArgs drgevent
);
```

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnEnabledChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnEnter(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnFontChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnForeColorChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnGiveFeedback(
   System.Windows.Forms.GiveFeedbackEventArgs gfbevent
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnGotFocus(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnHandleCreated(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnHandleDestroyed(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnHelpRequested(
   System.Windows.Forms.HelpEventArgs hevent
);
protected void OnImeModeChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnInvalidated(
   System.Windows.Forms.InvalidateEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnKeyDown(
   System.Windows.Forms.KeyEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnKeyPress(
   System.Windows.Forms.KeyPressEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnKeyUp(
   System.Windows.Forms.KeyEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnLayout(
   System.Windows.Forms.LayoutEventArgs levent
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnLeave(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnLocationChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
```

```
protected void OnLostFocus(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMouseDown(
   System.Windows.Forms.MouseEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMouseEnter(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMouseHover(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMouseLeave(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMouseMove(
   System.Windows.Forms.MouseEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMouseUp(
   System.Windows.Forms.MouseEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMouseWheel(
   System.Windows.Forms.MouseEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMove(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnNotifyMessage(
   System.Windows.Forms.Message m
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnPaint(
   System.Windows.Forms.PaintEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnPaintBackground(
   System.Windows.Forms.PaintEventArgs pevent
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnParentBackColorChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnParentBackgroundImageChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnParentBindingContextChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnParentChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnParentEnabledChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
```

```
protected void OnParentFontChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnParentForeColorChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnParentRightToLeftChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnParentVisibleChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnQueryContinueDrag(
   System.Windows.Forms.QueryContinueDragEventArgs qcdevent
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnResize(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnRightToLeftChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnSizeChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnStyleChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnSystemColorsChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnTabIndexChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnTabStopChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnTextChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnValidated(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnValidating(
   System.ComponentModel.CancelEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnVisibleChanged(
   System.EventArgs e
);
protected boolean ProcessCmdKey(
   System.Windows.Forms.Message& msg,
   System.Windows.Forms.Keys keyData
);
protected boolean ProcessDialogChar(
```

```
   char charCode
);
protected boolean ProcessDialogKey(
   System.Windows.Forms.Keys keyData
);
protected boolean ProcessKeyEventArgs(
   System.Windows.Forms.Message& m
);
protected boolean ProcessKeyMessage(
   System.Windows.Forms.Message& m
);
protected boolean ProcessKeyPreview(
   System.Windows.Forms.Message& m
);
protected boolean ProcessMnemonic(
   char charCode
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RaiseDragEvent(
   System.Object key,
   System.Windows.Forms.DragEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RaiseKeyEvent(
   System.Object key,
   System.Windows.Forms.KeyEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RaiseMouseEvent(
   System.Object key,
   System.Windows.Forms.MouseEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RaisePaintEvent(
   System.Object key,
   System.Windows.Forms.PaintEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void RecreateHandle();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected static boolean ReflectMessage(
   System.IntPtr hWnd,
   System.Windows.Forms.Message& m
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void ResetMouseEventArgs();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected System.Drawing.ContentAlignment RtlTranslateAlignment(
   System.Drawing.ContentAlignment align
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected System.Windows.Forms.HorizontalAlignment RtlTranslateAlignment(
   System.Windows.Forms.HorizontalAlignment align
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected System.Windows.Forms.LeftRightAlignment RtlTranslateAlignment(
   System.Windows.Forms.LeftRightAlignment align
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected System.Drawing.ContentAlignment RtlTranslateContent(
   System.Drawing.ContentAlignment align
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected System.Windows.Forms.HorizontalAlignment RtlTranslateHorizontal(
   System.Windows.Forms.HorizontalAlignment align
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected System.Windows.Forms.LeftRightAlignment RtlTranslateLeftRight(
```

```
   System.Windows.Forms.LeftRightAlignment align
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void ScaleCore(
   float dx,
   float dy
);
protected void Select(
   boolean directed,
   boolean forward
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void SetClientSizeCore(
   int x,
   int y
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void SetStyle(
   System.Windows.Forms.ControlStyles flag,
   boolean value
);
protected void SetTopLevel(
   boolean value
);
protected void SetVisibleCore(
   boolean value
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void UpdateBounds();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void UpdateBounds(
   int x,
   int y,
   int width,
   int height
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void UpdateBounds(
   int x,
   int y,
   int width,
   int height,
   int clientWidth,
   int clientHeight
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void UpdateStyles();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void UpdateZOrder();
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

Control.ControlAccessibleObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class ControlAccessibleObject
   extends System.Windows.Forms.AccessibleObject
```

Public Constructors

```
public ControlAccessibleObject(
   System.Windows.Forms.Control ownerControl
);
```

Public Properties

```
public System.String get_DefaultAction();
public System.String get_Description();
public System.IntPtr get_Handle();
public void set_Handle(
   System.IntPtr value
);
public System.String get_Help();
public System.String get_KeyboardShortcut();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Windows.Forms.Control get_Owner();
public System.Windows.Forms.AccessibleRole get_Role();
```

Public Methods

```
public int GetHelpTopic(
   System.String& fileName
);
public void NotifyClients(
   System.Windows.Forms.AccessibleEvents accEvent
);
public void NotifyClients(
   System.Windows.Forms.AccessibleEvents accEvent,
   int childID
);
public System.String ToString();
```

Control.ControlCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.ListBindableAttribute(ListBindable = false) */
public class ControlCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable, System.ICloneable
```

Public Constructors

```
public ControlCollection(
   System.Windows.Forms.Control owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public System.Windows.Forms.Control get_Item(
   int index
);
```

Public Methods

```
public void Add(
   System.Windows.Forms.Control value
);
public void AddRange(
   System.Windows.Forms.Control[] controls
);
public void Clear();
public boolean Contains(
   System.Windows.Forms.Control control
);
public final void CopyTo(
   System.Array dest,
   int index
);
public boolean Equals(
   System.Object other
);
public int GetChildIndex(
   System.Windows.Forms.Control child
);
public int GetChildIndex(
   System.Windows.Forms.Control child,
   boolean throwException
);
public final System.Collections.IEnumerator GetEnumerator();
public int GetHashCode();
public int IndexOf(
   System.Windows.Forms.Control control
);
public void Remove(
```

```
   System.Windows.Forms.Control value
);
public final void RemoveAt(
   int index
);
public void SetChildIndex(
   System.Windows.Forms.Control child,
   int newIndex
);
```

ControlBindingsCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "CollectionChanged") */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Windo
ws.Forms.Design.ControlBindingsConverter, System.Design, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Drawing.Design.UITypeEditor, System.Drawing, Version=1.0
.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class ControlBindingsCollection
   extends System.Windows.Forms.BindingsCollection
```

Public Properties

```
public System.Windows.Forms.Control get_Control();
public System.Windows.Forms.Binding get_Item(
   System.String propertyName
);
```

Public Methods

```
public void Add(
   System.Windows.Forms.Binding binding
);
public System.Windows.Forms.Binding Add(
   System.String propertyName,
   System.Object dataSource,
   System.String dataMember
);
public void Clear();
public void Remove(
   System.Windows.Forms.Binding binding
);
public void RemoveAt(
   int index
);
```

Protected Methods

```
protected void AddCore(
   System.Windows.Forms.Binding dataBinding
);
protected void ClearCore();
protected void RemoveCore(
   System.Windows.Forms.Binding dataBinding
);
```

ControlEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ControlEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ControlEventArgs(
   System.Windows.Forms.Control control
);
```

Public Properties

```
public System.Windows.Forms.Control get_Control();
```

ControlEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ControlEventHandler(
   System.Object sender,
   System.Windows.Forms.ControlEventArgs e
);
```

ControlPaint Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ControlPaint
   extends System.Object
```

Public Properties

```
public static System.Drawing.Color get_ContrastControlDark();
```

Public Methods

```
public static System.IntPtr CreateHBitmap16Bit(
   System.Drawing.Bitmap bitmap,
   System.Drawing.Color background
);
public static System.IntPtr CreateHBitmapColorMask(
   System.Drawing.Bitmap bitmap,
   System.IntPtr monochromeMask
);
public static System.IntPtr CreateHBitmapTransparencyMask(
   System.Drawing.Bitmap bitmap
);
public static System.Drawing.Color Dark(
   System.Drawing.Color baseColor
);
public static System.Drawing.Color Dark(
   System.Drawing.Color baseColor,
   float percOfDarkDark
);
public static System.Drawing.Color DarkDark(
   System.Drawing.Color baseColor
);
public static void DrawBorder(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle bounds,
   System.Drawing.Color color,
   System.Windows.Forms.ButtonBorderStyle style
);
public static void DrawBorder(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle bounds,
   System.Drawing.Color leftColor,
   int leftWidth,
   System.Windows.Forms.ButtonBorderStyle leftStyle,
   System.Drawing.Color topColor,
   int topWidth,
   System.Windows.Forms.ButtonBorderStyle topStyle,
   System.Drawing.Color rightColor,
   int rightWidth,
   System.Windows.Forms.ButtonBorderStyle rightStyle,
   System.Drawing.Color bottomColor,
   int bottomWidth,
   System.Windows.Forms.ButtonBorderStyle bottomStyle
);
public static void DrawBorder3D(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle rectangle
```

```
);
public static void DrawBorder3D(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle rectangle,
   System.Windows.Forms.Border3DStyle style
);
public static void DrawBorder3D(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle rectangle,
   System.Windows.Forms.Border3DStyle style,
   System.Windows.Forms.Border3DSide sides
);
public static void DrawBorder3D(
   System.Drawing.Graphics graphics,
   int x,
   int y,
   int width,
   int height
);
public static void DrawBorder3D(
   System.Drawing.Graphics graphics,
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.Border3DStyle style
);
public static void DrawBorder3D(
   System.Drawing.Graphics graphics,
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.Border3DStyle style,
   System.Windows.Forms.Border3DSide sides
);
public static void DrawButton(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle rectangle,
   System.Windows.Forms.ButtonState state
);
public static void DrawButton(
   System.Drawing.Graphics graphics,
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.ButtonState state
);
public static void DrawCaptionButton(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle rectangle,
   System.Windows.Forms.CaptionButton button,
   System.Windows.Forms.ButtonState state
);
public static void DrawCaptionButton(
   System.Drawing.Graphics graphics,
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.CaptionButton button,
   System.Windows.Forms.ButtonState state
);
public static void DrawCheckBox(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle rectangle,
   System.Windows.Forms.ButtonState state
```

```
);
public static void DrawCheckBox(
   System.Drawing.Graphics graphics,
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.ButtonState state
);
public static void DrawComboButton(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle rectangle,
   System.Windows.Forms.ButtonState state
);
public static void DrawComboButton(
   System.Drawing.Graphics graphics,
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.ButtonState state
);
public static void DrawContainerGrabHandle(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle bounds
);
public static void DrawFocusRectangle(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle rectangle
);
public static void DrawFocusRectangle(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle rectangle,
   System.Drawing.Color foreColor,
   System.Drawing.Color backColor
);
public static void DrawGrabHandle(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle rectangle,
   boolean primary,
   boolean enabled
);
public static void DrawGrid(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle area,
   System.Drawing.Size pixelsBetweenDots,
   System.Drawing.Color backColor
);
public static void DrawImageDisabled(
   System.Drawing.Graphics graphics,
   System.Drawing.Image image,
   int x,
   int y,
   System.Drawing.Color background
);
public static void DrawLockedFrame(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle rectangle,
   boolean primary
);
public static void DrawMenuGlyph(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle rectangle,
   System.Windows.Forms.MenuGlyph glyph
);
public static void DrawMenuGlyph(
   System.Drawing.Graphics graphics,
   int x,
```

```
      int y,
      int width,
      int height,
      System.Windows.Forms.MenuGlyph glyph
   );
   public static void DrawMixedCheckBox(
      System.Drawing.Graphics graphics,
      System.Drawing.Rectangle rectangle,
      System.Windows.Forms.ButtonState state
   );
   public static void DrawMixedCheckBox(
      System.Drawing.Graphics graphics,
      int x,
      int y,
      int width,
      int height,
      System.Windows.Forms.ButtonState state
   );
   public static void DrawRadioButton(
      System.Drawing.Graphics graphics,
      System.Drawing.Rectangle rectangle,
      System.Windows.Forms.ButtonState state
   );
   public static void DrawRadioButton(
      System.Drawing.Graphics graphics,
      int x,
      int y,
      int width,
      int height,
      System.Windows.Forms.ButtonState state
   );
   public static void DrawReversibleFrame(
      System.Drawing.Rectangle rectangle,
      System.Drawing.Color backColor,
      System.Windows.Forms.FrameStyle style
   );
   public static void DrawReversibleLine(
      System.Drawing.Point start,
      System.Drawing.Point end,
      System.Drawing.Color backColor
   );
   public static void DrawScrollButton(
      System.Drawing.Graphics graphics,
      System.Drawing.Rectangle rectangle,
      System.Windows.Forms.ScrollButton button,
      System.Windows.Forms.ButtonState state
   );
   public static void DrawScrollButton(
      System.Drawing.Graphics graphics,
      int x,
      int y,
      int width,
      int height,
      System.Windows.Forms.ScrollButton button,
      System.Windows.Forms.ButtonState state
   );
   public static void DrawSelectionFrame(
      System.Drawing.Graphics graphics,
      boolean active,
      System.Drawing.Rectangle outsideRect,
      System.Drawing.Rectangle insideRect,
      System.Drawing.Color backColor
   );
   public static void DrawSizeGrip(
      System.Drawing.Graphics graphics,
      System.Drawing.Color backColor,
      System.Drawing.Rectangle bounds
   );
```

```
public static void DrawSizeGrip(
   System.Drawing.Graphics graphics,
   System.Drawing.Color backColor,
   int x,
   int y,
   int width,
   int height
);
public static void DrawStringDisabled(
   System.Drawing.Graphics graphics,
   System.String s,
   System.Drawing.Font font,
   System.Drawing.Color color,
   System.Drawing.RectangleF layoutRectangle,
   System.Drawing.StringFormat format
);
public static void FillReversibleRectangle(
   System.Drawing.Rectangle rectangle,
   System.Drawing.Color backColor
);
public static System.Drawing.Color Light(
   System.Drawing.Color baseColor
);
public static System.Drawing.Color Light(
   System.Drawing.Color baseColor,
   float percOfLightLight
);
public static System.Drawing.Color LightLight(
   System.Drawing.Color baseColor
);
```

ControlStyles Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum ControlStyles
```

Members

AllPaintingInWmPaint

CacheText

ContainerControl

DoubleBuffer

EnableNotifyMessage

FixedHeight

FixedWidth

Opaque

ResizeRedraw

Selectable

StandardClick

StandardDoubleClick

SupportsTransparentBackColor

UserMouse

UserPaint

ConvertEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ConvertEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ConvertEventArgs(
   System.Object value,
   System.Type desiredType
);
```

Public Properties

```
public System.Type get_DesiredType();
public System.Object get_Value();
public void set_Value(
   System.Object value
);
```

ConvertEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ConvertEventHandler(
   System.Object sender,
   System.Windows.Forms.ConvertEventArgs e
);
```

CreateParams Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CreateParams
   extends System.Object
```

Public Constructors

```
public CreateParams();
```

Public Properties

```
public System.String get_Caption();
public void set_Caption(
   System.String value
);
public System.String get_ClassName();
public void set_ClassName(
   System.String value
);
public int get_ClassStyle();
public void set_ClassStyle(
   int value
);
public int get_ExStyle();
public void set_ExStyle(
   int value
);
public int get_Height();
public void set_Height(
   int value
);
public System.Object get_Param();
public void set_Param(
   System.Object value
);
public System.IntPtr get_Parent();
public void set_Parent(
   System.IntPtr value
);
public int get_Style();
public void set_Style(
   int value
);
public int get_Width();
public void set_Width(
   int value
);
public int get_X();
public void set_X(
   int value
);
public int get_Y();
public void set_Y(
   int value
);
```

Public Methods

```
public System.String ToString();
```

CurrencyManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CurrencyManager
   extends System.Windows.Forms.BindingManagerBase
```

Public Properties

```
public int get_Count();
public System.Object get_Current();
public System.Collections.IList get_List();
public int get_Position();
public void set_Position(
   int value
);
```

Public Methods

```
public void AddNew();
public void CancelCurrentEdit();
public void EndCurrentEdit();
public System.ComponentModel.PropertyDescriptorCollection GetItemProperties();
public void Refresh();
public void RemoveAt(
   int index
);
public void ResumeBinding();
public void SuspendBinding();
```

Public Events

```
public void add_ItemChanged(
   System.Windows.Forms.ItemChangedEventHandler value
);
public void remove_ItemChanged(
   System.Windows.Forms.ItemChangedEventHandler value
);
```

Protected Methods

```
protected void CheckEmpty();
protected System.String GetListName(
   System.Collections.ArrayList listAccessors
);
protected void OnCurrentChanged(
   System.EventArgs e
);
protected void OnItemChanged(
   System.Windows.Forms.ItemChangedEventArgs e
);
protected void OnPositionChanged(
   System.EventArgs e
);
```

```
protected void UpdateIsBinding();
```

Protected Fields

```
protected System.Type finalType;
protected int listposition;
```

Cursor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Drawing.Design.CursorEditor, System.Drawing.Design, Vers
ion=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Windo
ws.Forms.CursorConverter, System.Windows.Forms, Version=1.0.3300.0, Culture=neutral, PublicKe
yToken=b77a5c561934e089") */
public class Cursor
   extends System.Object
   implements System.IDisposable, System.Runtime.Serialization.ISerializable
```

Public Constructors

```
public Cursor(
   System.IntPtr handle
);
public Cursor(
   System.IO.Stream stream
);
public Cursor(
   System.String fileName
);
public Cursor(
   System.Type type,
   System.String resource
);
```

Public Properties

```
public static System.Drawing.Rectangle get_Clip();
public static void set_Clip(
   System.Drawing.Rectangle value
);
public static System.Windows.Forms.Cursor get_Current();
public static void set_Current(
   System.Windows.Forms.Cursor value
);
public System.IntPtr get_Handle();
public static System.Drawing.Point get_Position();
public static void set_Position(
   System.Drawing.Point value
);
public System.Drawing.Size get_Size();
```

Public Methods

```
public System.IntPtr CopyHandle();
public final void Dispose();
public void Draw(
   System.Drawing.Graphics g,
   System.Drawing.Rectangle targetRect
);
```

```
public void DrawStretched(
   System.Drawing.Graphics g,
   System.Drawing.Rectangle targetRect
);
public boolean Equals(
   System.Object obj
);
public int GetHashCode();
public static void Hide();
public static void Show();
public System.String ToString();
```

Public Operators

```
public static boolean op_Equality(
   System.Windows.Forms.Cursor left,
   System.Windows.Forms.Cursor right
);
public static boolean op_Inequality(
   System.Windows.Forms.Cursor left,
   System.Windows.Forms.Cursor right
);
```

Protected Methods

```
protected void Finalize();
```

CursorConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CursorConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public CursorConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

Cursors Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Cursors
   extends System.Object
```

Public Properties

```
public static System.Windows.Forms.Cursor get_AppStarting();
public static System.Windows.Forms.Cursor get_Arrow();
public static System.Windows.Forms.Cursor get_Cross();
public static System.Windows.Forms.Cursor get_Default();
public static System.Windows.Forms.Cursor get_Hand();
public static System.Windows.Forms.Cursor get_Help();
public static System.Windows.Forms.Cursor get_HSplit();
public static System.Windows.Forms.Cursor get_IBeam();
public static System.Windows.Forms.Cursor get_No();
public static System.Windows.Forms.Cursor get_NoMove2D();
public static System.Windows.Forms.Cursor get_NoMoveHoriz();
public static System.Windows.Forms.Cursor get_NoMoveVert();
public static System.Windows.Forms.Cursor get_PanEast();
public static System.Windows.Forms.Cursor get_PanNE();
public static System.Windows.Forms.Cursor get_PanNorth();
public static System.Windows.Forms.Cursor get_PanNW();
public static System.Windows.Forms.Cursor get_PanSE();
public static System.Windows.Forms.Cursor get_PanSouth();
public static System.Windows.Forms.Cursor get_PanSW();
public static System.Windows.Forms.Cursor get_PanWest();
public static System.Windows.Forms.Cursor get_SizeAll();
public static System.Windows.Forms.Cursor get_SizeNESW();
public static System.Windows.Forms.Cursor get_SizeNS();
public static System.Windows.Forms.Cursor get_SizeNWSE();
public static System.Windows.Forms.Cursor get_SizeWE();
public static System.Windows.Forms.Cursor get_UpArrow();
public static System.Windows.Forms.Cursor get_VSplit();
public static System.Windows.Forms.Cursor get_WaitCursor();
```

DataFormats Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataFormats
   extends System.Object
```

Public Methods

```
public static System.Windows.Forms.DataFormats+Format GetFormat(
   int id
);
public static System.Windows.Forms.DataFormats+Format GetFormat(
   System.String format
);
```

Public Fields

```
public static final System.String Bitmap;
public static final System.String CommaSeparatedValue;
public static final System.String Dib;
public static final System.String Dif;
public static final System.String EnhancedMetafile;
public static final System.String FileDrop;
public static final System.String Html;
public static final System.String Locale;
public static final System.String MetafilePict;
public static final System.String OemText;
public static final System.String Palette;
public static final System.String PenData;
public static final System.String Riff;
public static final System.String Rtf;
public static final System.String Serializable;
public static final System.String StringFormat;
public static final System.String SymbolicLink;
public static final System.String Text;
public static final System.String Tiff;
public static final System.String UnicodeText;
public static final System.String WaveAudio;
```

DataFormats.Format Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Format
   extends System.Object
```

Public Constructors

```
public Format(
   System.String name,
   int id
);
```

Public Properties

```
public int get_Id();
public System.String get_Name();
```

DataGrid Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Navigate") */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.DataGridDesigner,
System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "DataSource") */
public class DataGrid
   extends System.Windows.Forms.Control
   implements System.ComponentModel.ISupportInitialize, System.Windows.Forms.IDataGridEditing
Service
```

Public Constructors

```
public DataGrid();
```

Public Properties

```
public boolean get_AllowNavigation();
public void set_AllowNavigation(
   boolean value
);
public boolean get_AllowSorting();
public void set_AllowSorting(
   boolean value
);
public System.Drawing.Color get_AlternatingBackColor();
public void set_AlternatingBackColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_BackgroundColor();
public void set_BackgroundColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Windows.Forms.BorderStyle value
);
public System.Drawing.Color get_CaptionBackColor();
public void set_CaptionBackColor(
   System.Drawing.Color value
);
public System.Drawing.Font get_CaptionFont();
public void set_CaptionFont(
   System.Drawing.Font value
);
public System.Drawing.Color get_CaptionForeColor();
public void set_CaptionForeColor(
```

```
   System.Drawing.Color value
);
public System.String get_CaptionText();
public void set_CaptionText(
   System.String value
);
public boolean get_CaptionVisible();
public void set_CaptionVisible(
   boolean value
);
public boolean get_ColumnHeadersVisible();
public void set_ColumnHeadersVisible(
   boolean value
);
public System.Windows.Forms.DataGridCell get_CurrentCell();
public void set_CurrentCell(
   System.Windows.Forms.DataGridCell value
);
public int get_CurrentRowIndex();
public void set_CurrentRowIndex(
   int value
);
public System.Windows.Forms.Cursor get_Cursor();
public void set_Cursor(
   System.Windows.Forms.Cursor value
);
public System.String get_DataMember();
public void set_DataMember(
   System.String value
);
public System.Object get_DataSource();
public void set_DataSource(
   System.Object value
);
public int get_FirstVisibleColumn();
public boolean get_FlatMode();
public void set_FlatMode(
   boolean value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_GridLineColor();
public void set_GridLineColor(
   System.Drawing.Color value
);
public System.Windows.Forms.DataGridLineStyle get_GridLineStyle();
public void set_GridLineStyle(
   System.Windows.Forms.DataGridLineStyle value
);
public System.Drawing.Color get_HeaderBackColor();
public void set_HeaderBackColor(
   System.Drawing.Color value
);
public System.Drawing.Font get_HeaderFont();
public void set_HeaderFont(
   System.Drawing.Font value
);
public System.Drawing.Color get_HeaderForeColor();
public void set_HeaderForeColor(
   System.Drawing.Color value
);
public System.Object get_Item(
   System.Windows.Forms.DataGridCell cell
);
public System.Object get_Item(
   int rowIndex,
```

```
      int columnIndex
   );
   public void set_Item(
      System.Windows.Forms.DataGridCell cell,
      System.Object value
   );
   public void set_Item(
      int rowIndex,
      int columnIndex,
      System.Object value
   );
   public System.Drawing.Color get_LinkColor();
   public void set_LinkColor(
      System.Drawing.Color value
   );
   public System.Drawing.Color get_LinkHoverColor();
   public void set_LinkHoverColor(
      System.Drawing.Color value
   );
   public System.Drawing.Color get_ParentRowsBackColor();
   public void set_ParentRowsBackColor(
      System.Drawing.Color value
   );
   public System.Drawing.Color get_ParentRowsForeColor();
   public void set_ParentRowsForeColor(
      System.Drawing.Color value
   );
   public System.Windows.Forms.DataGridParentRowsLabelStyle get_ParentRowsLabelStyle();
   public void set_ParentRowsLabelStyle(
      System.Windows.Forms.DataGridParentRowsLabelStyle value
   );
   public boolean get_ParentRowsVisible();
   public void set_ParentRowsVisible(
      boolean value
   );
   public int get_PreferredColumnWidth();
   public void set_PreferredColumnWidth(
      int value
   );
   public int get_PreferredRowHeight();
   public void set_PreferredRowHeight(
      int value
   );
   public boolean get_ReadOnly();
   public void set_ReadOnly(
      boolean value
   );
   public boolean get_RowHeadersVisible();
   public void set_RowHeadersVisible(
      boolean value
   );
   public int get_RowHeaderWidth();
   public void set_RowHeaderWidth(
      int value
   );
   public System.Drawing.Color get_SelectionBackColor();
   public void set_SelectionBackColor(
      System.Drawing.Color value
   );
   public System.Drawing.Color get_SelectionForeColor();
   public void set_SelectionForeColor(
      System.Drawing.Color value
   );
   public System.ComponentModel.ISite get_Site();
   public void set_Site(
      System.ComponentModel.ISite value
   );
   public System.Windows.Forms.GridTableStylesCollection get_TableStyles();
```

```
public System.String get_Text();
public void set_Text(
   System.String value
);
public int get_VisibleColumnCount();
public int get_VisibleRowCount();
```

Public Methods

```
public final boolean BeginEdit(
   System.Windows.Forms.DataGridColumnStyle gridColumn,
   int rowNumber
);
public final void BeginInit();
public void Collapse(
   int row
);
public final boolean EndEdit(
   System.Windows.Forms.DataGridColumnStyle gridColumn,
   int rowNumber,
   boolean shouldAbort
);
public final void EndInit();
public void Expand(
   int row
);
public System.Drawing.Rectangle GetCellBounds(
   System.Windows.Forms.DataGridCell dgc
);
public System.Drawing.Rectangle GetCellBounds(
   int row,
   int col
);
public System.Drawing.Rectangle GetCurrentCellBounds();
public System.Windows.Forms.DataGrid+HitTestInfo HitTest(
   System.Drawing.Point position
);
public System.Windows.Forms.DataGrid+HitTestInfo HitTest(
   int x,
   int y
);
public boolean IsExpanded(
   int rowNumber
);
public boolean IsSelected(
   int row
);
public void NavigateBack();
public void NavigateTo(
   int rowNumber,
   System.String relationName
);
public void ResetAlternatingBackColor();
public void ResetBackColor();
public void ResetForeColor();
public void ResetGridLineColor();
public void ResetHeaderBackColor();
public void ResetHeaderFont();
public void ResetHeaderForeColor();
public void ResetLinkColor();
public void ResetLinkHoverColor();
public void ResetSelectionBackColor();
public void ResetSelectionForeColor();
public void Select(
   int row
);
```

```
public void SetDataBinding(
   System.Object dataSource,
   System.String dataMember
);
public void SubObjectsSiteChange(
   boolean site
);
public void UnSelect(
   int row
);
```

Public Events

```
public void add_AllowNavigationChanged(
   System.EventHandler value
);
public void remove_AllowNavigationChanged(
   System.EventHandler value
);
public void add_BackButtonClick(
   System.EventHandler value
);
public void remove_BackButtonClick(
   System.EventHandler value
);
public void add_BackgroundColorChanged(
   System.EventHandler value
);
public void remove_BackgroundColorChanged(
   System.EventHandler value
);
public void add_BorderStyleChanged(
   System.EventHandler value
);
public void remove_BorderStyleChanged(
   System.EventHandler value
);
public void add_CaptionVisibleChanged(
   System.EventHandler value
);
public void remove_CaptionVisibleChanged(
   System.EventHandler value
);
public void add_CurrentCellChanged(
   System.EventHandler value
);
public void remove_CurrentCellChanged(
   System.EventHandler value
);
public void add_DataSourceChanged(
   System.EventHandler value
);
public void remove_DataSourceChanged(
   System.EventHandler value
);
public void add_FlatModeChanged(
   System.EventHandler value
);
public void remove_FlatModeChanged(
   System.EventHandler value
);
public void add_Navigate(
   System.Windows.Forms.NavigateEventHandler value
);
public void remove_Navigate(
   System.Windows.Forms.NavigateEventHandler value
```

```
);
public void add_ParentRowsLabelStyleChanged(
   System.EventHandler value
);
public void remove_ParentRowsLabelStyleChanged(
   System.EventHandler value
);
public void add_ParentRowsVisibleChanged(
   System.EventHandler value
);
public void remove_ParentRowsVisibleChanged(
   System.EventHandler value
);
public void add_ReadOnlyChanged(
   System.EventHandler value
);
public void remove_ReadOnlyChanged(
   System.EventHandler value
);
public void add_Scroll(
   System.EventHandler value
);
public void remove_Scroll(
   System.EventHandler value
);
public void add_ShowParentDetailsButtonClick(
   System.EventHandler value
);
public void remove_ShowParentDetailsButtonClick(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Drawing.Size get_DefaultSize();
protected System.Windows.Forms.ScrollBar get_HorizScrollBar();
protected System.Windows.Forms.CurrencyManager get_ListManager();
protected void set_ListManager(
   System.Windows.Forms.CurrencyManager value
);
protected System.Windows.Forms.ScrollBar get_VertScrollBar();
```

Protected Methods

```
protected void CancelEditing();
protected void ColumnStartedEditing(
   System.Drawing.Rectangle bounds
);
protected void ColumnStartedEditing(
   System.Windows.Forms.Control editingControl
);
protected System.Windows.Forms.AccessibleObject CreateAccessibilityInstance();
protected System.Windows.Forms.DataGridColumnStyle CreateGridColumn(
   System.ComponentModel.PropertyDescriptor prop
);
protected System.Windows.Forms.DataGridColumnStyle CreateGridColumn(
   System.ComponentModel.PropertyDescriptor prop,
   boolean isDefault
);
protected void Dispose(
   boolean disposing
);
protected System.String GetOutputTextDelimiter();
protected void GridHScrolled(
```

```
   System.Object sender,
   System.Windows.Forms.ScrollEventArgs se
);
protected void GridVScrolled(
   System.Object sender,
   System.Windows.Forms.ScrollEventArgs se
);
protected void OnAllowNavigationChanged(
   System.EventArgs e
);
protected void OnBackButtonClicked(
   System.Object sender,
   System.EventArgs e
);
protected void OnBackColorChanged(
   System.EventArgs e
);
protected void OnBackgroundColorChanged(
   System.EventArgs e
);
protected void OnBindingContextChanged(
   System.EventArgs e
);
protected void OnBorderStyleChanged(
   System.EventArgs e
);
protected void OnCaptionVisibleChanged(
   System.EventArgs e
);
protected void OnCurrentCellChanged(
   System.EventArgs e
);
protected void OnDataSourceChanged(
   System.EventArgs e
);
protected void OnEnter(
   System.EventArgs e
);
protected void OnFlatModeChanged(
   System.EventArgs e
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnForeColorChanged(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnHandleDestroyed(
   System.EventArgs e
);
protected void OnKeyDown(
   System.Windows.Forms.KeyEventArgs ke
);
protected void OnKeyPress(
   System.Windows.Forms.KeyPressEventArgs kpe
);
protected void OnLayout(
   System.Windows.Forms.LayoutEventArgs levent
);
protected void OnLeave(
   System.EventArgs e
);
protected void OnMouseDown(
   System.Windows.Forms.MouseEventArgs e
);
```

```
protected void OnMouseLeave(
   System.EventArgs e
);
protected void OnMouseMove(
   System.Windows.Forms.MouseEventArgs e
);
protected void OnMouseUp(
   System.Windows.Forms.MouseEventArgs e
);
protected void OnMouseWheel(
   System.Windows.Forms.MouseEventArgs e
);
protected void OnNavigate(
   System.Windows.Forms.NavigateEventArgs e
);
protected void OnPaint(
   System.Windows.Forms.PaintEventArgs pe
);
protected void OnPaintBackground(
   System.Windows.Forms.PaintEventArgs ebe
);
protected void OnParentRowsLabelStyleChanged(
   System.EventArgs e
);
protected void OnParentRowsVisibleChanged(
   System.EventArgs e
);
protected void OnReadOnlyChanged(
   System.EventArgs e
);
protected void OnResize(
   System.EventArgs e
);
protected void OnRowHeaderClick(
   System.EventArgs e
);
protected void OnScroll(
   System.EventArgs e
);
protected void OnShowParentDetailsButtonClicked(
   System.Object sender,
   System.EventArgs e
);
protected boolean ProcessDialogKey(
   System.Windows.Forms.Keys keyData
);
protected boolean ProcessGridKey(
   System.Windows.Forms.KeyEventArgs ke
);
protected boolean ProcessKeyPreview(
   System.Windows.Forms.Message& m
);
protected boolean ProcessTabKey(
   System.Windows.Forms.Keys keyData
);
protected void ResetSelection();
protected boolean ShouldSerializeAlternatingBackColor();
protected boolean ShouldSerializeBackgroundColor();
protected boolean ShouldSerializeCaptionBackColor();
protected boolean ShouldSerializeCaptionForeColor();
protected boolean ShouldSerializeGridLineColor();
protected boolean ShouldSerializeHeaderBackColor();
protected boolean ShouldSerializeHeaderFont();
protected boolean ShouldSerializeHeaderForeColor();
protected boolean ShouldSerializeLinkHoverColor();
protected boolean ShouldSerializeParentRowsBackColor();
protected boolean ShouldSerializeParentRowsForeColor();
protected boolean ShouldSerializePreferredRowHeight();
```

```
protected boolean ShouldSerializeSelectionBackColor();
protected boolean ShouldSerializeSelectionForeColor();
```

Protected Events

```
protected void add_RowHeaderClick(
   System.EventHandler value
);
protected void remove_RowHeaderClick(
   System.EventHandler value
);
```

DataGrid.HitTestInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HitTestInfo
   extends System.Object
```

Public Properties

```
public int get_Column();
public int get_Row();
public System.Windows.Forms.DataGrid+HitTestType get_Type();
```

Public Methods

```
public boolean Equals(
   System.Object value
);
public int GetHashCode();
public System.String ToString();
```

Public Fields

```
public static final System.Windows.Forms.DataGrid+HitTestInfo Nowhere;
```

DataGrid.HitTestType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum HitTestType
```

Members

Caption

Cell

ColumnHeader

ColumnResize

None

ParentRows

RowHeader

RowResize

DataGridBoolColumn Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataGridBoolColumn
   extends System.Windows.Forms.DataGridColumnStyle
```

Public Constructors

```
public DataGridBoolColumn();
public DataGridBoolColumn(
   System.ComponentModel.PropertyDescriptor prop
);
public DataGridBoolColumn(
   System.ComponentModel.PropertyDescriptor prop,
   boolean isDefault
);
```

Public Properties

```
public boolean get_AllowNull();
public void set_AllowNull(
   boolean value
);
public System.Object get_FalseValue();
public void set_FalseValue(
   System.Object value
);
public System.Object get_NullValue();
public void set_NullValue(
   System.Object value
);
public System.Object get_TrueValue();
public void set_TrueValue(
   System.Object value
);
```

Public Events

```
public void add_AllowNullChanged(
   System.EventHandler value
);
public void remove_AllowNullChanged(
   System.EventHandler value
);
public void add_FalseValueChanged(
   System.EventHandler value
);
public void remove_FalseValueChanged(
   System.EventHandler value
);
public void add_TrueValueChanged(
   System.EventHandler value
);
public void remove_TrueValueChanged(
   System.EventHandler value
);
```

Protected Methods

```
protected void Abort(
   int rowNum
);
protected boolean Commit(
   System.Windows.Forms.CurrencyManager dataSource,
   int rowNum
);
protected void ConcedeFocus();
protected void Edit(
   System.Windows.Forms.CurrencyManager source,
   int rowNum,
   System.Drawing.Rectangle bounds,
   boolean readOnly,
   System.String instantText,
   boolean cellIsVisible
);
protected void EnterNullValue();
protected System.Object GetColumnValueAtRow(
   System.Windows.Forms.CurrencyManager lm,
   int row
);
protected int GetMinimumHeight();
protected int GetPreferredHeight(
   System.Drawing.Graphics g,
   System.Object value
);
protected System.Drawing.Size GetPreferredSize(
   System.Drawing.Graphics g,
   System.Object value
);
protected void Paint(
   System.Drawing.Graphics g,
   System.Drawing.Rectangle bounds,
   System.Windows.Forms.CurrencyManager source,
   int rowNum
);
protected void Paint(
   System.Drawing.Graphics g,
   System.Drawing.Rectangle bounds,
   System.Windows.Forms.CurrencyManager source,
   int rowNum,
   boolean alignToRight
);
protected void Paint(
   System.Drawing.Graphics g,
   System.Drawing.Rectangle bounds,
   System.Windows.Forms.CurrencyManager source,
   int rowNum,
   System.Drawing.Brush backBrush,
   System.Drawing.Brush foreBrush,
   boolean alignToRight
);
protected void SetColumnValueAtRow(
   System.Windows.Forms.CurrencyManager lm,
   int row,
   System.Object value
);
```

DataGridCell Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataGridCell
   extends System.ValueType
```

Public Constructors

```
public DataGridCell(
   int r,
   int c
);
```

Public Properties

```
public int get_ColumnNumber();
public void set_ColumnNumber(
   int value
);
public int get_RowNumber();
public void set_RowNumber(
   int value
);
```

Public Methods

```
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public System.String ToString();
```

DataGridColumnStyle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Header") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
public abstract class DataGridColumnStyle
   extends System.ComponentModel.Component
   implements System.Windows.Forms.IDataGridColumnStyleEditingNotificationService
```

Public Constructors

```
public DataGridColumnStyle();
public DataGridColumnStyle(
   System.ComponentModel.PropertyDescriptor prop
);
```

Public Properties

```
public System.Windows.Forms.HorizontalAlignment get_Alignment();
public void set_Alignment(
   System.Windows.Forms.HorizontalAlignment value
);
public System.Windows.Forms.DataGridTableStyle get_DataGridTableStyle();
public System.Windows.Forms.AccessibleObject get_HeaderAccessibleObject();
public System.String get_HeaderText();
public void set_HeaderText(
   System.String value
);
public System.String get_MappingName();
public void set_MappingName(
   System.String value
);
public System.String get_NullText();
public void set_NullText(
   System.String value
);
public System.ComponentModel.PropertyDescriptor get_PropertyDescriptor();
public void set_PropertyDescriptor(
   System.ComponentModel.PropertyDescriptor value
);
public boolean get_ReadOnly();
public void set_ReadOnly(
   boolean value
);
public int get_Width();
public void set_Width(
   int value
);
```

Public Methods

```
public void ResetHeaderText();
```

Public Events

```
public void add_AlignmentChanged(
   System.EventHandler value
);
public void remove_AlignmentChanged(
   System.EventHandler value
);
public void add_FontChanged(
   System.EventHandler value
);
public void remove_FontChanged(
   System.EventHandler value
);
public void add_HeaderTextChanged(
   System.EventHandler value
);
public void remove_HeaderTextChanged(
   System.EventHandler value
);
public void add_MappingNameChanged(
   System.EventHandler value
);
public void remove_MappingNameChanged(
   System.EventHandler value
);
public void add_NullTextChanged(
   System.EventHandler value
);
public void remove_NullTextChanged(
   System.EventHandler value
);
public void add_PropertyDescriptorChanged(
   System.EventHandler value
);
public void remove_PropertyDescriptorChanged(
   System.EventHandler value
);
public void add_ReadOnlyChanged(
   System.EventHandler value
);
public void remove_ReadOnlyChanged(
   System.EventHandler value
);
public void add_WidthChanged(
   System.EventHandler value
);
public void remove_WidthChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected int get_FontHeight();
```

Protected Methods

```
protected abstract void Abort(
   int rowNum
);
protected void BeginUpdate();
protected void CheckValidDataSource(
   System.Windows.Forms.CurrencyManager value
```

```
);
protected void ColumnStartedEditing(
   System.Windows.Forms.Control editingControl
);
protected abstract boolean Commit(
   System.Windows.Forms.CurrencyManager dataSource,
   int rowNum
);
protected void ConcedeFocus();
protected System.Windows.Forms.AccessibleObject CreateHeaderAccessibleObject();
protected void Edit(
   System.Windows.Forms.CurrencyManager source,
   int rowNum,
   System.Drawing.Rectangle bounds,
   boolean readOnly
);
protected void Edit(
   System.Windows.Forms.CurrencyManager source,
   int rowNum,
   System.Drawing.Rectangle bounds,
   boolean readOnly,
   System.String instantText
);
protected abstract void Edit(
   System.Windows.Forms.CurrencyManager source,
   int rowNum,
   System.Drawing.Rectangle bounds,
   boolean readOnly,
   System.String instantText,
   boolean cellIsVisible
);
protected void EndUpdate();
protected void EnterNullValue();
protected System.Object GetColumnValueAtRow(
   System.Windows.Forms.CurrencyManager source,
   int rowNum
);
protected abstract int GetMinimumHeight();
protected abstract int GetPreferredHeight(
   System.Drawing.Graphics g,
   System.Object value
);
protected abstract System.Drawing.Size GetPreferredSize(
   System.Drawing.Graphics g,
   System.Object value
);
protected void Invalidate();
protected abstract void Paint(
   System.Drawing.Graphics g,
   System.Drawing.Rectangle bounds,
   System.Windows.Forms.CurrencyManager source,
   int rowNum
);
protected abstract void Paint(
   System.Drawing.Graphics g,
   System.Drawing.Rectangle bounds,
   System.Windows.Forms.CurrencyManager source,
   int rowNum,
   boolean alignToRight
);
protected void Paint(
   System.Drawing.Graphics g,
   System.Drawing.Rectangle bounds,
   System.Windows.Forms.CurrencyManager source,
   int rowNum,
   System.Drawing.Brush backBrush,
   System.Drawing.Brush foreBrush,
   boolean alignToRight
```

```
);
protected void SetColumnValueAtRow(
   System.Windows.Forms.CurrencyManager source,
   int rowNum,
   System.Object value
);
protected void SetDataGrid(
   System.Windows.Forms.DataGrid value
);
protected void SetDataGridInColumn(
   System.Windows.Forms.DataGrid value
);
protected void UpdateUI(
   System.Windows.Forms.CurrencyManager source,
   int rowNum,
   System.String instantText
);
```

DataGridColumnStyle.CompModSwitches Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
protected class CompModSwitches
   extends System.Object
```

Public Constructors

```
public CompModSwitches();
```

Public Properties

```
public static System.Diagnostics.TraceSwitch get_DGEditColumnEditing();
```

DataGridColumnStyle.DataGridColumnHeaderAccessibleObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
protected class DataGridColumnHeaderAccessibleObject
   extends System.Windows.Forms.AccessibleObject
```

Public Constructors

```
public DataGridColumnHeaderAccessibleObject(
   System.Windows.Forms.DataGridColumnStyle owner
);
```

Public Properties

```
public System.Drawing.Rectangle get_Bounds();
public System.String get_Name();
public System.Windows.Forms.AccessibleObject get_Parent();
public System.Windows.Forms.AccessibleRole get_Role();
```

Public Methods

```
public System.Windows.Forms.AccessibleObject Navigate(
   System.Windows.Forms.AccessibleNavigation navdir
);
```

Protected Properties

```
protected System.Windows.Forms.DataGridColumnStyle get_Owner();
```

DataGridLineStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum DataGridLineStyle
```

Members

None

Solid

DataGridParentRowsLabelStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum DataGridParentRowsLabelStyle
```

Members

Both

ColumnName

None

TableName

DataGridPreferredColumnWidthTypeConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataGridPreferredColumnWidthTypeConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public DataGridPreferredColumnWidthTypeConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

DataGridTableStyle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class DataGridTableStyle
   extends System.ComponentModel.Component
   implements System.Windows.Forms.IDataGridEditingService
```

Public Constructors

```
public DataGridTableStyle();
public DataGridTableStyle(
   boolean isDefaultTableStyle
);
public DataGridTableStyle(
   System.Windows.Forms.CurrencyManager listManager
);
```

Public Properties

```
public boolean get_AllowSorting();
public void set_AllowSorting(
   boolean value
);
public System.Drawing.Color get_AlternatingBackColor();
public void set_AlternatingBackColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public boolean get_ColumnHeadersVisible();
public void set_ColumnHeadersVisible(
   boolean value
);
public System.Windows.Forms.DataGrid get_DataGrid();
public void set_DataGrid(
   System.Windows.Forms.DataGrid value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.Windows.Forms.GridColumnStylesCollection get_GridColumnStyles();
public System.Drawing.Color get_GridLineColor();
public void set_GridLineColor(
   System.Drawing.Color value
);
public System.Windows.Forms.DataGridLineStyle get_GridLineStyle();
public void set_GridLineStyle(
   System.Windows.Forms.DataGridLineStyle value
);
public System.Drawing.Color get_HeaderBackColor();
public void set_HeaderBackColor(
   System.Drawing.Color value
);
```

```
public System.Drawing.Font get_HeaderFont();
public void set_HeaderFont(
   System.Drawing.Font value
);
public System.Drawing.Color get_HeaderForeColor();
public void set_HeaderForeColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_LinkColor();
public void set_LinkColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_LinkHoverColor();
public void set_LinkHoverColor(
   System.Drawing.Color value
);
public System.String get_MappingName();
public void set_MappingName(
   System.String value
);
public int get_PreferredColumnWidth();
public void set_PreferredColumnWidth(
   int value
);
public int get_PreferredRowHeight();
public void set_PreferredRowHeight(
   int value
);
public boolean get_ReadOnly();
public void set_ReadOnly(
   boolean value
);
public boolean get_RowHeadersVisible();
public void set_RowHeadersVisible(
   boolean value
);
public int get_RowHeaderWidth();
public void set_RowHeaderWidth(
   int value
);
public System.Drawing.Color get_SelectionBackColor();
public void set_SelectionBackColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_SelectionForeColor();
public void set_SelectionForeColor(
   System.Drawing.Color value
);
```

Public Methods

```
public final boolean BeginEdit(
   System.Windows.Forms.DataGridColumnStyle gridColumn,
   int rowNumber
);
public final boolean EndEdit(
   System.Windows.Forms.DataGridColumnStyle gridColumn,
   int rowNumber,
   boolean shouldAbort
);
public void ResetAlternatingBackColor();
public void ResetBackColor();
public void ResetForeColor();
public void ResetGridLineColor();
public void ResetHeaderBackColor();
public void ResetHeaderFont();
```

```
public void ResetHeaderForeColor();
public void ResetLinkColor();
public void ResetLinkHoverColor();
public void ResetSelectionBackColor();
public void ResetSelectionForeColor();
```

Public Events

```
public void add_AllowSortingChanged(
   System.EventHandler value
);
public void remove_AllowSortingChanged(
   System.EventHandler value
);
public void add_AlternatingBackColorChanged(
   System.EventHandler value
);
public void remove_AlternatingBackColorChanged(
   System.EventHandler value
);
public void add_BackColorChanged(
   System.EventHandler value
);
public void remove_BackColorChanged(
   System.EventHandler value
);
public void add_ColumnHeadersVisibleChanged(
   System.EventHandler value
);
public void remove_ColumnHeadersVisibleChanged(
   System.EventHandler value
);
public void add_ForeColorChanged(
   System.EventHandler value
);
public void remove_ForeColorChanged(
   System.EventHandler value
);
public void add_GridLineColorChanged(
   System.EventHandler value
);
public void remove_GridLineColorChanged(
   System.EventHandler value
);
public void add_GridLineStyleChanged(
   System.EventHandler value
);
public void remove_GridLineStyleChanged(
   System.EventHandler value
);
public void add_HeaderBackColorChanged(
   System.EventHandler value
);
public void remove_HeaderBackColorChanged(
   System.EventHandler value
);
public void add_HeaderFontChanged(
   System.EventHandler value
);
public void remove_HeaderFontChanged(
   System.EventHandler value
);
public void add_HeaderForeColorChanged(
   System.EventHandler value
);
public void remove_HeaderForeColorChanged(
```

```
    System.EventHandler value
);
public void add_LinkColorChanged(
    System.EventHandler value
);
public void remove_LinkColorChanged(
    System.EventHandler value
);
public void add_LinkHoverColorChanged(
    System.EventHandler value
);
public void remove_LinkHoverColorChanged(
    System.EventHandler value
);
public void add_MappingNameChanged(
    System.EventHandler value
);
public void remove_MappingNameChanged(
    System.EventHandler value
);
public void add_PreferredColumnWidthChanged(
    System.EventHandler value
);
public void remove_PreferredColumnWidthChanged(
    System.EventHandler value
);
public void add_PreferredRowHeightChanged(
    System.EventHandler value
);
public void remove_PreferredRowHeightChanged(
    System.EventHandler value
);
public void add_ReadOnlyChanged(
    System.EventHandler value
);
public void remove_ReadOnlyChanged(
    System.EventHandler value
);
public void add_RowHeadersVisibleChanged(
    System.EventHandler value
);
public void remove_RowHeadersVisibleChanged(
    System.EventHandler value
);
public void add_RowHeaderWidthChanged(
    System.EventHandler value
);
public void remove_RowHeaderWidthChanged(
    System.EventHandler value
);
public void add_SelectionBackColorChanged(
    System.EventHandler value
);
public void remove_SelectionBackColorChanged(
    System.EventHandler value
);
public void add_SelectionForeColorChanged(
    System.EventHandler value
);
public void remove_SelectionForeColorChanged(
    System.EventHandler value
);
```

Public Fields

```
public static System.Windows.Forms.DataGridTableStyle DefaultTableStyle;
```

Protected Methods

```
protected System.Windows.Forms.DataGridColumnStyle CreateGridColumn(
   System.ComponentModel.PropertyDescriptor prop
);
protected System.Windows.Forms.DataGridColumnStyle CreateGridColumn(
   System.ComponentModel.PropertyDescriptor prop,
   boolean isDefault
);
protected void Dispose(
   boolean disposing
);
protected void OnAllowSortingChanged(
   System.EventArgs e
);
protected void OnAlternatingBackColorChanged(
   System.EventArgs e
);
protected void OnBackColorChanged(
   System.EventArgs e
);
protected void OnColumnHeadersVisibleChanged(
   System.EventArgs e
);
protected void OnForeColorChanged(
   System.EventArgs e
);
protected void OnGridLineColorChanged(
   System.EventArgs e
);
protected void OnGridLineStyleChanged(
   System.EventArgs e
);
protected void OnHeaderBackColorChanged(
   System.EventArgs e
);
protected void OnHeaderFontChanged(
   System.EventArgs e
);
protected void OnHeaderForeColorChanged(
   System.EventArgs e
);
protected void OnLinkColorChanged(
   System.EventArgs e
);
protected void OnLinkHoverColorChanged(
   System.EventArgs e
);
protected void OnMappingNameChanged(
   System.EventArgs e
);
protected void OnPreferredColumnWidthChanged(
   System.EventArgs e
);
protected void OnPreferredRowHeightChanged(
   System.EventArgs e
);
protected void OnReadOnlyChanged(
   System.EventArgs e
);
protected void OnRowHeadersVisibleChanged(
   System.EventArgs e
);
protected void OnRowHeaderWidthChanged(
   System.EventArgs e
);
```

```
protected void OnSelectionBackColorChanged(
   System.EventArgs e
);
protected void OnSelectionForeColorChanged(
   System.EventArgs e
);
protected boolean ShouldSerializeAlternatingBackColor();
protected boolean ShouldSerializeBackColor();
protected boolean ShouldSerializeForeColor();
protected boolean ShouldSerializeGridLineColor();
protected boolean ShouldSerializeHeaderBackColor();
protected boolean ShouldSerializeHeaderForeColor();
protected boolean ShouldSerializeLinkColor();
protected boolean ShouldSerializeLinkHoverColor();
protected boolean ShouldSerializePreferredRowHeight();
protected boolean ShouldSerializeSelectionBackColor();
protected boolean ShouldSerializeSelectionForeColor();
```

DataGridTextBox Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "GridEditName") */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
public class DataGridTextBox
   extends System.Windows.Forms.TextBox
```

Public Constructors

```
public DataGridTextBox();
```

Public Properties

```
public boolean get_IsInEditOrNavigateMode();
public void set_IsInEditOrNavigateMode(
   boolean value
);
```

Public Methods

```
public void SetDataGrid(
   System.Windows.Forms.DataGrid parentGrid
);
```

Protected Methods

```
protected void OnKeyPress(
   System.Windows.Forms.KeyPressEventArgs e
);
protected void OnMouseWheel(
   System.Windows.Forms.MouseEventArgs e
);
protected boolean ProcessKeyMessage(
   System.Windows.Forms.Message& m
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

DataGridTextBoxColumn Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DataGridTextBoxColumn
   extends System.Windows.Forms.DataGridColumnStyle
```

Public Constructors

```
public DataGridTextBoxColumn();
public DataGridTextBoxColumn(
   System.ComponentModel.PropertyDescriptor prop
);
public DataGridTextBoxColumn(
   System.ComponentModel.PropertyDescriptor prop,
   boolean isDefault
);
public DataGridTextBoxColumn(
   System.ComponentModel.PropertyDescriptor prop,
   System.String format
);
public DataGridTextBoxColumn(
   System.ComponentModel.PropertyDescriptor prop,
   System.String format,
   boolean isDefault
);
```

Public Properties

```
public System.String get_Format();
public void set_Format(
   System.String value
);
public System.IFormatProvider get_FormatInfo();
public void set_FormatInfo(
   System.IFormatProvider value
);
public void set_PropertyDescriptor(
   System.ComponentModel.PropertyDescriptor value
);
public boolean get_ReadOnly();
public void set_ReadOnly(
   boolean value
);
public System.Windows.Forms.TextBox get_TextBox();
```

Protected Methods

```
protected void Abort(
   int rowNum
);
protected boolean Commit(
   System.Windows.Forms.CurrencyManager dataSource,
   int rowNum
);
protected void ConcedeFocus();
protected void Edit(
```

```
   System.Windows.Forms.CurrencyManager source,
   int rowNum,
   System.Drawing.Rectangle bounds,
   boolean readOnly,
   System.String instantText,
   boolean cellIsVisible
);
protected void EndEdit();
protected void EnterNullValue();
protected int GetMinimumHeight();
protected int GetPreferredHeight(
   System.Drawing.Graphics g,
   System.Object value
);
protected System.Drawing.Size GetPreferredSize(
   System.Drawing.Graphics g,
   System.Object value
);
protected void HideEditBox();
protected void Paint(
   System.Drawing.Graphics g,
   System.Drawing.Rectangle bounds,
   System.Windows.Forms.CurrencyManager source,
   int rowNum
);
protected void Paint(
   System.Drawing.Graphics g,
   System.Drawing.Rectangle bounds,
   System.Windows.Forms.CurrencyManager source,
   int rowNum,
   boolean alignToRight
);
protected void Paint(
   System.Drawing.Graphics g,
   System.Drawing.Rectangle bounds,
   System.Windows.Forms.CurrencyManager source,
   int rowNum,
   System.Drawing.Brush backBrush,
   System.Drawing.Brush foreBrush,
   boolean alignToRight
);
protected void PaintText(
   System.Drawing.Graphics g,
   System.Drawing.Rectangle bounds,
   System.String text,
   boolean alignToRight
);
protected void PaintText(
   System.Drawing.Graphics g,
   System.Drawing.Rectangle textBounds,
   System.String text,
   System.Drawing.Brush backBrush,
   System.Drawing.Brush foreBrush,
   boolean alignToRight
);
protected void SetDataGridInColumn(
   System.Windows.Forms.DataGrid value
);
protected void UpdateUI(
   System.Windows.Forms.CurrencyManager source,
   int rowNum,
   System.String instantText
);
```

DataObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ClassInterfaceAttribute(Value = None) */
public class DataObject
   extends System.Object
   implements System.Windows.Forms.IDataObject, System.Windows.Forms.UnsafeNativeMethods+IOle
DataObject
```

Public Constructors

```
public DataObject();
public DataObject(
   System.Object data
);
public DataObject(
   System.String format,
   System.Object data
);
```

Public Methods

```
public System.Object GetData(
   System.String format
);
public System.Object GetData(
   System.Type format
);
public System.Object GetData(
   System.String format,
   boolean autoConvert
);
public boolean GetDataPresent(
   System.String format
);
public boolean GetDataPresent(
   System.Type format
);
public boolean GetDataPresent(
   System.String format,
   boolean autoConvert
);
public System.String[] GetFormats();
public System.String[] GetFormats(
   boolean autoConvert
);
public void SetData(
   System.Object data
);
public void SetData(
   System.String format,
   System.Object data
);
public void SetData(
   System.Type format,
   System.Object data
);
public void SetData(
```

```
    System.String format,
    boolean autoConvert,
    System.Object data
);
```

DateBoldEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DateBoldEventArgs
   extends System.EventArgs
```

Public Properties

```
public int[] get_DaysToBold();
public void set_DaysToBold(
   int[] value
);
public int get_Size();
public System.DateTime get_StartDate();
```

DateBoldEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DateBoldEventHandler(
   System.Object sender,
   System.Windows.Forms.DateBoldEventArgs e
);
```

DateRangeEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DateRangeEventArgs
   extends System.EventArgs
```

Public Constructors

```
public DateRangeEventArgs(
   System.DateTime start,
   System.DateTime end
);
```

Public Properties

```
public System.DateTime get_End();
public System.DateTime get_Start();
```

DateRangeEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DateRangeEventHandler(
   System.Object sender,
   System.Windows.Forms.DateRangeEventArgs e
);
```

DateTimePicker Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Value") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ValueChanged") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.DateTimePickerDesi
gner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") *
/
public class DateTimePicker
   extends System.Windows.Forms.Control
```

Public Constructors

```
public DateTimePicker();
```

Public Properties

```
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Drawing.Font get_CalendarFont();
public void set_CalendarFont(
   System.Drawing.Font value
);
public System.Drawing.Color get_CalendarForeColor();
public void set_CalendarForeColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_CalendarMonthBackground();
public void set_CalendarMonthBackground(
   System.Drawing.Color value
);
public System.Drawing.Color get_CalendarTitleBackColor();
public void set_CalendarTitleBackColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_CalendarTitleForeColor();
public void set_CalendarTitleForeColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_CalendarTrailingForeColor();
public void set_CalendarTrailingForeColor(
   System.Drawing.Color value
);
public boolean get_Checked();
public void set_Checked(
   boolean value
);
public System.String get_CustomFormat();
public void set_CustomFormat(
   System.String value
);
```

```
public System.Windows.Forms.LeftRightAlignment get_DropDownAlign();
public void set_DropDownAlign(
   System.Windows.Forms.LeftRightAlignment value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.Windows.Forms.DateTimePickerFormat get_Format();
public void set_Format(
   System.Windows.Forms.DateTimePickerFormat value
);
public System.DateTime get_MaxDate();
public void set_MaxDate(
   System.DateTime value
);
public System.DateTime get_MinDate();
public void set_MinDate(
   System.DateTime value
);
public int get_PreferredHeight();
public boolean get_ShowCheckBox();
public void set_ShowCheckBox(
   boolean value
);
public boolean get_ShowUpDown();
public void set_ShowUpDown(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.DateTime get_Value();
public void set_Value(
   System.DateTime value
);
```

Public Methods

```
public System.String ToString();
```

Public Events

```
public void add_CloseUp(
   System.EventHandler value
);
public void remove_CloseUp(
   System.EventHandler value
);
public void add_DropDown(
   System.EventHandler value
);
public void remove_DropDown(
   System.EventHandler value
);
public void add_FormatChanged(
   System.EventHandler value
);
public void remove_FormatChanged(
   System.EventHandler value
);
public void add_Paint(
   System.Windows.Forms.PaintEventHandler value
```

```
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void add_ValueChanged(
   System.EventHandler value
);
public void remove_ValueChanged(
   System.EventHandler value
);
```

Public Fields

```
public static final System.DateTime MaxDateTime;
public static final System.DateTime MinDateTime;
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected System.Windows.Forms.AccessibleObject CreateAccessibilityInstance();
protected void CreateHandle();
protected void DestroyHandle();
protected boolean IsInputKey(
   System.Windows.Forms.Keys keyData
);
protected void OnCloseUp(
   System.EventArgs eventargs
);
protected void OnDropDown(
   System.EventArgs eventargs
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnFormatChanged(
   System.EventArgs e
);
protected void OnSystemColorsChanged(
   System.EventArgs e
);
protected void OnValueChanged(
   System.EventArgs eventargs
);
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

Protected Fields

```
protected static final System.Drawing.Color DefaultMonthBackColor;
protected static final System.Drawing.Color DefaultTitleBackColor;
protected static final System.Drawing.Color DefaultTitleForeColor;
protected static final System.Drawing.Color DefaultTrailingForeColor;
```

DateTimePicker.DateTimePickerAccessibleObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class DateTimePickerAccessibleObject
   extends System.Windows.Forms.Control+ControlAccessibleObject
```

Public Constructors

```
public DateTimePickerAccessibleObject(
   System.Windows.Forms.DateTimePicker owner
);
```

Public Properties

```
public System.Windows.Forms.AccessibleStates get_State();
public System.String get_Value();
```

DateTimePickerFormat Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum DateTimePickerFormat
```

Members

Custom

Long

Short

Time

Day Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum Day
```

Members

Default

Friday

Monday

Saturday

Sunday

Thursday

Tuesday

Wednesday

DialogResult Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum DialogResult
```

Members

Abort

Cancel

Ignore

No

None

OK

Retry

Yes

DockStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Windows.Forms.Design.DockEditor, System.Design, Version=
1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public enum DockStyle
```

Members

Bottom

Fill

Left

None

Right

Top

DomainUpDown Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Items") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "SelectedItemChanged") */
public class DomainUpDown
   extends System.Windows.Forms.UpDownBase
```

Public Constructors

```
public DomainUpDown();
```

Public Properties

```
public System.Windows.Forms.DomainUpDown+DomainUpDownItemCollection get_Items();
public int get_SelectedIndex();
public void set_SelectedIndex(
   int value
);
public System.Object get_SelectedItem();
public void set_SelectedItem(
   System.Object value
);
public boolean get_Sorted();
public void set_Sorted(
   boolean value
);
public boolean get_Wrap();
public void set_Wrap(
   boolean value
);
```

Public Methods

```
public void DownButton();
public System.String ToString();
public void UpButton();
```

Public Events

```
public void add_SelectedItemChanged(
   System.EventHandler value
);
public void remove_SelectedItemChanged(
   System.EventHandler value
);
```

Protected Methods

```
protected System.Windows.Forms.AccessibleObject CreateAccessibilityInstance();
protected void OnChanged(
   System.Object source,
```

```
   System.EventArgs e
);
protected void OnSelectedItemChanged(
   System.Object source,
   System.EventArgs e
);
protected void OnTextBoxKeyDown(
   System.Object source,
   System.Windows.Forms.KeyEventArgs e
);
protected void UpdateEditText();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

DomainUpDown.DomainItemAccessibleObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class DomainItemAccessibleObject
   extends System.Windows.Forms.AccessibleObject
```

Public Constructors

```
public DomainItemAccessibleObject(
   System.String name,
   System.Windows.Forms.AccessibleObject parent
);
```

Public Properties

```
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Windows.Forms.AccessibleObject get_Parent();
public System.Windows.Forms.AccessibleRole get_Role();
public System.Windows.Forms.AccessibleStates get_State();
public System.String get_Value();
```

DomainUpDown.DomainUpDownAccessibleObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class DomainUpDownAccessibleObject
   extends System.Windows.Forms.Control+ControlAccessibleObject
```

Public Constructors

```
public DomainUpDownAccessibleObject(
   System.Windows.Forms.Control owner
);
```

Public Properties

```
public System.Windows.Forms.AccessibleRole get_Role();
```

Public Methods

```
public System.Windows.Forms.AccessibleObject GetChild(
   int index
);
public int GetChildCount();
```

DomainUpDown.DomainUpDownItemCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class DomainUpDownItemCollection
   extends System.Collections.ArrayList
```

Public Properties

```
public System.Object get_Item(
   int index
);
public void set_Item(
   int index,
   System.Object value
);
```

Public Methods

```
public int Add(
   System.Object item
);
public void Insert(
   int index,
   System.Object item
);
public void Remove(
   System.Object item
);
public void RemoveAt(
   int item
);
```

DragAction Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum DragAction
```

Members

Cancel

Continue

Drop

DragDropEffects Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum DragDropEffects
```

Members

All

Copy

Link

Move

None

Scroll

DragEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class DragEventArgs
   extends System.EventArgs
```

Public Constructors

```
public DragEventArgs(
   System.Windows.Forms.IDataObject data,
   int keyState,
   int x,
   int y,
   System.Windows.Forms.DragDropEffects allowedEffect,
   System.Windows.Forms.DragDropEffects effect
);
```

Public Properties

```
public System.Windows.Forms.DragDropEffects get_AllowedEffect();
public System.Windows.Forms.IDataObject get_Data();
public System.Windows.Forms.DragDropEffects get_Effect();
public void set_Effect(
   System.Windows.Forms.DragDropEffects value
);
public int get_KeyState();
public int get_X();
public int get_Y();
```

DragEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DragEventHandler(
   System.Object sender,
   System.Windows.Forms.DragEventArgs e
);
```

DrawItemEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DrawItemEventArgs
   extends System.EventArgs
```

Public Constructors

```
public DrawItemEventArgs(
   System.Drawing.Graphics graphics,
   System.Drawing.Font font,
   System.Drawing.Rectangle rect,
   int index,
   System.Windows.Forms.DrawItemState state
);
public DrawItemEventArgs(
   System.Drawing.Graphics graphics,
   System.Drawing.Font font,
   System.Drawing.Rectangle rect,
   int index,
   System.Windows.Forms.DrawItemState state,
   System.Drawing.Color foreColor,
   System.Drawing.Color backColor
);
```

Public Properties

```
public System.Drawing.Color get_BackColor();
public System.Drawing.Rectangle get_Bounds();
public System.Drawing.Font get_Font();
public System.Drawing.Color get_ForeColor();
public System.Drawing.Graphics get_Graphics();
public int get_Index();
public System.Windows.Forms.DrawItemState get_State();
```

Public Methods

```
public void DrawBackground();
public void DrawFocusRectangle();
```

DrawItemEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void DrawItemEventHandler(
   System.Object sender,
   System.Windows.Forms.DrawItemEventArgs e
);
```

DrawItemState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum DrawItemState
```

Members

Checked

ComboBoxEdit

Default

Disabled

Focus

Grayed

HotLight

Inactive

NoAccelerator

NoFocusRect

None

Selected

DrawMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum DrawMode
```

Members

Normal

OwnerDrawFixed

OwnerDrawVariable

ErrorBlinkStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ErrorBlinkStyle
```

Members

AlwaysBlink

BlinkIfDifferentError

NeverBlink

ErrorIconAlignment Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ErrorIconAlignment
```

Members

BottomLeft

BottomRight

MiddleLeft

MiddleRight

TopLeft

TopRight

ErrorProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemFilterAttribute(FilterString = "System.Window
s.Forms", FilterType = Allow) */
/** @attribute System.ComponentModel.ProvidePropertyAttribute(PropertyName = "Error", Receive
rTypeName = "System.Windows.Forms.Control, System.Windows.Forms, Version=1.0.3300.0, Culture=
neutral, PublicKeyToken=b77a5c561934e089") */
/** @attribute System.ComponentModel.ProvidePropertyAttribute(PropertyName = "IconPadding", R
eceiverTypeName = "System.Windows.Forms.Control, System.Windows.Forms, Version=1.0.3300.0, Cu
lture=neutral, PublicKeyToken=b77a5c561934e089") */
/** @attribute System.ComponentModel.ProvidePropertyAttribute(PropertyName = "IconAlignment",
ReceiverTypeName = "System.Windows.Forms.Control, System.Windows.Forms, Version=1.0.3300.0, C
ulture=neutral, PublicKeyToken=b77a5c561934e089") */
public class ErrorProvider
   extends System.ComponentModel.Component
   implements System.ComponentModel.IExtenderProvider
```

Public Constructors

```
public ErrorProvider();
public ErrorProvider(
   System.Windows.Forms.ContainerControl parentControl
);
```

Public Properties

```
public int get_BlinkRate();
public void set_BlinkRate(
   int value
);
public System.Windows.Forms.ErrorBlinkStyle get_BlinkStyle();
public void set_BlinkStyle(
   System.Windows.Forms.ErrorBlinkStyle value
);
public System.Windows.Forms.ContainerControl get_ContainerControl();
public void set_ContainerControl(
   System.Windows.Forms.ContainerControl value
);
public System.String get_DataMember();
public void set_DataMember(
   System.String value
);
public System.Object get_DataSource();
public void set_DataSource(
   System.Object value
);
public System.Drawing.Icon get_Icon();
public void set_Icon(
   System.Drawing.Icon value
);
public void set_Site(
   System.ComponentModel.ISite value
);
```

Public Methods

```
public void BindToDataAndErrors(
   System.Object newDataSource,
   System.String newDataMember
);
public final boolean CanExtend(
   System.Object extendee
);
/** @attribute System.Windows.Forms.SRDescriptionAttribute(Description = "The error descripti
on for a control.") */
/** @attribute System.Windows.Forms.SRCategoryAttribute(Category = "Appearance") */
/** @attribute System.ComponentModel.DefaultValueAttribute(Value = ) */
/** @attribute System.ComponentModel.LocalizableAttribute(IsLocalizable = true) */
public System.String GetError(
   System.Windows.Forms.Control control
);
/** @attribute System.Windows.Forms.SRDescriptionAttribute(Description = "The location of the
error icon relative to the control.") */
/** @attribute System.ComponentModel.DefaultValueAttribute(Value = MiddleRight) */
/** @attribute System.ComponentModel.LocalizableAttribute(IsLocalizable = true) */
/** @attribute System.Windows.Forms.SRCategoryAttribute(Category = "Appearance") */
public System.Windows.Forms.ErrorIconAlignment GetIconAlignment(
   System.Windows.Forms.Control control
);
/** @attribute System.Windows.Forms.SRDescriptionAttribute(Description = "The number of pixel
s to leave between the error icon and the control.") */
/** @attribute System.ComponentModel.DefaultValueAttribute(Value = 0) */
/** @attribute System.ComponentModel.LocalizableAttribute(IsLocalizable = true) */
/** @attribute System.Windows.Forms.SRCategoryAttribute(Category = "Appearance") */
public int GetIconPadding(
   System.Windows.Forms.Control control
);
public void SetError(
   System.Windows.Forms.Control control,
   System.String value
);
public void SetIconAlignment(
   System.Windows.Forms.Control control,
   System.Windows.Forms.ErrorIconAlignment value
);
public void SetIconPadding(
   System.Windows.Forms.Control control,
   int padding
);
public void UpdateBinding();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

FeatureSupport Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class FeatureSupport
   extends System.Object
   implements System.Windows.Forms.IFeatureSupport
```

Public Methods

```
public abstract System.Version GetVersionPresent(
   System.Object feature
);
public static System.Version GetVersionPresent(
   System.String featureClassName,
   System.String featureConstName
);
public boolean IsPresent(
   System.Object feature
);
public boolean IsPresent(
   System.Object feature,
   System.Version minimumVersion
);
public static boolean IsPresent(
   System.String featureClassName,
   System.String featureConstName
);
public static boolean IsPresent(
   System.String featureClassName,
   System.String featureConstName,
   System.Version minimumVersion
);
```

Protected Constructors

```
protected FeatureSupport();
```

FileDialog Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "FileName") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "FileOk") */
public abstract class FileDialog
   extends System.Windows.Forms.CommonDialog
```

Public Properties

```
public boolean get_AddExtension();
public void set_AddExtension(
   boolean value
);
public boolean get_CheckFileExists();
public void set_CheckFileExists(
   boolean value
);
public boolean get_CheckPathExists();
public void set_CheckPathExists(
   boolean value
);
public System.String get_DefaultExt();
public void set_DefaultExt(
   System.String value
);
public boolean get_DereferenceLinks();
public void set_DereferenceLinks(
   boolean value
);
public System.String get_FileName();
public void set_FileName(
   System.String value
);
public System.String[] get_FileNames();
public System.String get_Filter();
public void set_Filter(
   System.String value
);
public int get_FilterIndex();
public void set_FilterIndex(
   int value
);
public System.String get_InitialDirectory();
public void set_InitialDirectory(
   System.String value
);
public boolean get_RestoreDirectory();
public void set_RestoreDirectory(
   boolean value
);
public boolean get_ShowHelp();
public void set_ShowHelp(
   boolean value
);
public System.String get_Title();
public void set_Title(
   System.String value
);
public boolean get_ValidateNames();
```

```
public void set_ValidateNames(
   boolean value
);
```

Public Methods

```
public void Reset();
public System.String ToString();
```

Public Events

```
public void add_FileOk(
   System.ComponentModel.CancelEventHandler value
);
public void remove_FileOk(
   System.ComponentModel.CancelEventHandler value
);
```

Protected Properties

```
protected System.IntPtr get_Instance();
protected int get_Options();
```

Protected Methods

```
protected System.IntPtr HookProc(
   System.IntPtr hWnd,
   int msg,
   System.IntPtr wparam,
   System.IntPtr lparam
);
protected void OnFileOk(
   System.ComponentModel.CancelEventArgs e
);
protected boolean RunDialog(
   System.IntPtr hWndOwner
);
```

Protected Fields

```
protected static final System.Object EventFileOk;
```

FlatStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum FlatStyle
```

Members

Flat

Popup

Standard

System

FontDialog Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Apply") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Font") */
public class FontDialog
   extends System.Windows.Forms.CommonDialog
```

Public Constructors

```
public FontDialog();
```

Public Properties

```
public boolean get_AllowScriptChange();
public void set_AllowScriptChange(
   boolean value
);
public boolean get_AllowSimulations();
public void set_AllowSimulations(
   boolean value
);
public boolean get_AllowVectorFonts();
public void set_AllowVectorFonts(
   boolean value
);
public boolean get_AllowVerticalFonts();
public void set_AllowVerticalFonts(
   boolean value
);
public System.Drawing.Color get_Color();
public void set_Color(
   System.Drawing.Color value
);
public boolean get_FixedPitchOnly();
public void set_FixedPitchOnly(
   boolean value
);
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public boolean get_FontMustExist();
public void set_FontMustExist(
   boolean value
);
public int get_MaxSize();
public void set_MaxSize(
   int value
);
public int get_MinSize();
public void set_MinSize(
   int value
);
public boolean get_ScriptsOnly();
public void set_ScriptsOnly(
   boolean value
);
```

```
public boolean get_ShowApply();
public void set_ShowApply(
   boolean value
);
public boolean get_ShowColor();
public void set_ShowColor(
   boolean value
);
public boolean get_ShowEffects();
public void set_ShowEffects(
   boolean value
);
public boolean get_ShowHelp();
public void set_ShowHelp(
   boolean value
);
```

Public Methods

```
public void Reset();
public System.String ToString();
```

Public Events

```
public void add_Apply(
   System.EventHandler value
);
public void remove_Apply(
   System.EventHandler value
);
```

Protected Properties

```
protected int get_Options();
```

Protected Methods

```
protected System.IntPtr HookProc(
   System.IntPtr hWnd,
   int msg,
   System.IntPtr wparam,
   System.IntPtr lparam
);
protected void OnApply(
   System.EventArgs e
);
protected boolean RunDialog(
   System.IntPtr hWndOwner
);
```

Protected Fields

```
protected static final System.Object EventApply;
```

Form Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
/** @attribute System.ComponentModel.DesignerCategoryAttribute(Category = "Form") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Load") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IRootDesigner, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77
a5c561934e089", DesignerTypeName = "System.Windows.Forms.Design.FormDocumentDesigner, System.
Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class Form
   extends System.Windows.Forms.ContainerControl
```

Public Constructors

```
public Form();
```

Public Properties

```
public System.Windows.Forms.IButtonControl get_AcceptButton();
public void set_AcceptButton(
   System.Windows.Forms.IButtonControl value
);
public static System.Windows.Forms.Form get_ActiveForm();
public System.Windows.Forms.Form get_ActiveMdiChild();
public boolean get_AllowTransparency();
public void set_AllowTransparency(
   boolean value
);
public boolean get_AutoScale();
public void set_AutoScale(
   boolean value
);
public System.Drawing.Size get_AutoScaleBaseSize();
public void set_AutoScaleBaseSize(
   System.Drawing.Size value
);
public boolean get_AutoScroll();
public void set_AutoScroll(
   boolean value
);
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Windows.Forms.IButtonControl get_CancelButton();
public void set_CancelButton(
   System.Windows.Forms.IButtonControl value
);
public System.Drawing.Size get_ClientSize();
public void set_ClientSize(
   System.Drawing.Size value
);
public boolean get_ControlBox();
public void set_ControlBox(
   boolean value
);
```

```
public System.Drawing.Rectangle get_DesktopBounds();
public void set_DesktopBounds(
   System.Drawing.Rectangle value
);
public System.Drawing.Point get_DesktopLocation();
public void set_DesktopLocation(
   System.Drawing.Point value
);
public System.Windows.Forms.DialogResult get_DialogResult();
public void set_DialogResult(
   System.Windows.Forms.DialogResult value
);
public System.Windows.Forms.FormBorderStyle get_FormBorderStyle();
public void set_FormBorderStyle(
   System.Windows.Forms.FormBorderStyle value
);
public boolean get_HelpButton();
public void set_HelpButton(
   boolean value
);
public System.Drawing.Icon get_Icon();
public void set_Icon(
   System.Drawing.Icon value
);
public boolean get_IsMdiChild();
public boolean get_IsMdiContainer();
public void set_IsMdiContainer(
   boolean value
);
public boolean get_IsRestrictedWindow();
public boolean get_KeyPreview();
public void set_KeyPreview(
   boolean value
);
public boolean get_MaximizeBox();
public void set_MaximizeBox(
   boolean value
);
public System.Drawing.Size get_MaximumSize();
public void set_MaximumSize(
   System.Drawing.Size value
);
public System.Windows.Forms.Form[] get_MdiChildren();
public System.Windows.Forms.Form get_MdiParent();
public void set_MdiParent(
   System.Windows.Forms.Form value
);
public System.Windows.Forms.MainMenu get_Menu();
public void set_Menu(
   System.Windows.Forms.MainMenu value
);
public System.Windows.Forms.MainMenu get_MergedMenu();
public boolean get_MinimizeBox();
public void set_MinimizeBox(
   boolean value
);
public System.Drawing.Size get_MinimumSize();
public void set_MinimumSize(
   System.Drawing.Size value
);
public boolean get_Modal();
public double get_Opacity();
public void set_Opacity(
   double value
);
public System.Windows.Forms.Form[] get_OwnedForms();
public System.Windows.Forms.Form get_Owner();
public void set_Owner(
```

```
   System.Windows.Forms.Form value
);
public boolean get_ShowInTaskbar();
public void set_ShowInTaskbar(
   boolean value
);
public System.Drawing.Size get_Size();
public void set_Size(
   System.Drawing.Size value
);
public System.Windows.Forms.SizeGripStyle get_SizeGripStyle();
public void set_SizeGripStyle(
   System.Windows.Forms.SizeGripStyle value
);
public System.Windows.Forms.FormStartPosition get_StartPosition();
public void set_StartPosition(
   System.Windows.Forms.FormStartPosition value
);
public int get_TabIndex();
public void set_TabIndex(
   int value
);
public boolean get_TopLevel();
public void set_TopLevel(
   boolean value
);
public boolean get_TopMost();
public void set_TopMost(
   boolean value
);
public System.Drawing.Color get_TransparencyKey();
public void set_TransparencyKey(
   System.Drawing.Color value
);
public System.Windows.Forms.FormWindowState get_WindowState();
public void set_WindowState(
   System.Windows.Forms.FormWindowState value
);
```

Public Methods

```
public void Activate();
public void AddOwnedForm(
   System.Windows.Forms.Form ownedForm
);
public void Close();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
public static System.Drawing.SizeF GetAutoScaleSize(
   System.Drawing.Font font
);
public void LayoutMdi(
   System.Windows.Forms.MdiLayout value
);
public void RemoveOwnedForm(
   System.Windows.Forms.Form ownedForm
);
public void SetDesktopBounds(
   int x,
   int y,
   int width,
   int height
);
public void SetDesktopLocation(
   int x,
   int y
);
```

```
public System.Windows.Forms.DialogResult ShowDialog();
public System.Windows.Forms.DialogResult ShowDialog(
   System.Windows.Forms.IWin32Window owner
);
public System.String ToString();
```

Public Events

```
public void add_Activated(
   System.EventHandler value
);
public void remove_Activated(
   System.EventHandler value
);
public void add_Closed(
   System.EventHandler value
);
public void remove_Closed(
   System.EventHandler value
);
public void add_Closing(
   System.ComponentModel.CancelEventHandler value
);
public void remove_Closing(
   System.ComponentModel.CancelEventHandler value
);
public void add_Deactivate(
   System.EventHandler value
);
public void remove_Deactivate(
   System.EventHandler value
);
public void add_InputLanguageChanged(
   System.Windows.Forms.InputLanguageChangedEventHandler value
);
public void remove_InputLanguageChanged(
   System.Windows.Forms.InputLanguageChangedEventHandler value
);
public void add_InputLanguageChanging(
   System.Windows.Forms.InputLanguageChangingEventHandler value
);
public void remove_InputLanguageChanging(
   System.Windows.Forms.InputLanguageChangingEventHandler value
);
public void add_Load(
   System.EventHandler value
);
public void remove_Load(
   System.EventHandler value
);
public void add_MaximizedBoundsChanged(
   System.EventHandler value
);
public void remove_MaximizedBoundsChanged(
   System.EventHandler value
);
public void add_MaximumSizeChanged(
   System.EventHandler value
);
public void remove_MaximumSizeChanged(
   System.EventHandler value
);
public void add_MdiChildActivate(
   System.EventHandler value
);
public void remove_MdiChildActivate(
```

```
   System.EventHandler value
);
public void add_MenuComplete(
   System.EventHandler value
);
public void remove_MenuComplete(
   System.EventHandler value
);
public void add_MenuStart(
   System.EventHandler value
);
public void remove_MenuStart(
   System.EventHandler value
);
public void add_MinimumSizeChanged(
   System.EventHandler value
);
public void remove_MinimumSizeChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Windows.Forms.ImeMode get_DefaultImeMode();
protected System.Drawing.Size get_DefaultSize();
protected System.Drawing.Rectangle get_MaximizedBounds();
protected void set_MaximizedBounds(
   System.Drawing.Rectangle value
);
```

Protected Methods

```
protected void ActivateMdiChild(
   System.Windows.Forms.Form form
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void AdjustFormScrollbars(
   boolean displayScrollbars
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void ApplyAutoScaling();
protected void CenterToParent();
protected void CenterToScreen();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected System.Windows.Forms.Control+ControlCollection CreateControlsInstance();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void CreateHandle();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void DefWndProc(
   System.Windows.Forms.Message& m
);
protected void Dispose(
   boolean disposing
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnActivated(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnClosed(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
```

```
protected void OnClosing(
   System.ComponentModel.CancelEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnCreateControl();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnDeactivate(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnFontChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnHandleCreated(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnHandleDestroyed(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnInputLanguageChanged(
   System.Windows.Forms.InputLanguageChangedEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnInputLanguageChanging(
   System.Windows.Forms.InputLanguageChangingEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnLoad(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMaximizedBoundsChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMaximumSizeChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMdiChildActivate(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMenuComplete(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMenuStart(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMinimumSizeChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnPaint(
   System.Windows.Forms.PaintEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnResize(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnStyleChanged(
   System.EventArgs e
```

```
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnTextChanged(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnVisibleChanged(
   System.EventArgs e
);
protected boolean ProcessCmdKey(
   System.Windows.Forms.Message& msg,
   System.Windows.Forms.Keys keyData
);
protected boolean ProcessDialogKey(
   System.Windows.Forms.Keys keyData
);
protected boolean ProcessKeyPreview(
   System.Windows.Forms.Message& m
);
protected boolean ProcessTabKey(
   boolean forward
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void ScaleCore(
   float x,
   float y
);
protected void Select(
   boolean directed,
   boolean forward
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void SetClientSizeCore(
   int x,
   int y
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void SetVisibleCore(
   boolean value
);
protected void UpdateDefaultButton();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

Form.ControlCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ControlCollection
   extends System.Windows.Forms.Control+ControlCollection
```

Public Constructors

```
public ControlCollection(
   System.Windows.Forms.Form owner
);
```

Public Methods

```
public void Add(
   System.Windows.Forms.Control value
);
public void Remove(
   System.Windows.Forms.Control value
);
```

FormBorderStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum FormBorderStyle
```

Members

Fixed3D

FixedDialog

FixedSingle

FixedToolWindow

None

Sizable

SizableToolWindow

FormStartPosition Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum FormStartPosition
```

Members

CenterParent

CenterScreen

Manual

WindowsDefaultBounds

WindowsDefaultLocation

FormWindowState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum FormWindowState
```

Members

Maximized

Minimized

Normal

FrameStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum FrameStyle
```

Members

Dashed

Thick

GiveFeedbackEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class GiveFeedbackEventArgs
   extends System.EventArgs
```

Public Constructors

```
public GiveFeedbackEventArgs(
   System.Windows.Forms.DragDropEffects effect,
   boolean useDefaultCursors
);
```

Public Properties

```
public System.Windows.Forms.DragDropEffects get_Effect();
public boolean get_UseDefaultCursors();
public void set_UseDefaultCursors(
   boolean value
);
```

GiveFeedbackEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void GiveFeedbackEventHandler(
   System.Object sender,
   System.Windows.Forms.GiveFeedbackEventArgs e
);
```

GridColumnStylesCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Windows.Forms.Design.DataGridColumnCollectionEditor, Sys
tem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.ListBindableAttribute(ListBindable = false) */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class GridColumnStylesCollection
   extends System.Windows.Forms.BaseCollection
   implements System.Collections.IList
```

Public Properties

```
public System.Windows.Forms.DataGridColumnStyle get_Item(
   int index
);
public System.Windows.Forms.DataGridColumnStyle get_Item(
   System.ComponentModel.PropertyDescriptor propDesc
);
public System.Windows.Forms.DataGridColumnStyle get_Item(
   System.String columnName
);
```

Public Methods

```
public int Add(
   System.Windows.Forms.DataGridColumnStyle column
);
public void AddRange(
   System.Windows.Forms.DataGridColumnStyle[] columns
);
public void Clear();
public boolean Contains(
   System.ComponentModel.PropertyDescriptor propDesc
);
public boolean Contains(
   System.String name
);
public boolean Contains(
   System.Windows.Forms.DataGridColumnStyle column
);
public int IndexOf(
   System.Windows.Forms.DataGridColumnStyle element
);
public void Remove(
   System.Windows.Forms.DataGridColumnStyle column
);
public void RemoveAt(
   int index
);
public void ResetPropertyDescriptors();
```

Public Events

```
public void add_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
public void remove_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
```

Protected Properties

```
protected System.Collections.ArrayList get_List();
```

Protected Methods

```
protected void OnCollectionChanged(
   System.ComponentModel.CollectionChangeEventArgs ccevent
);
```

GridItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class GridItem
   extends System.Object
```

Public Properties

```
public boolean get_Expandable();
public boolean get_Expanded();
public void set_Expanded(
   boolean value
);
public abstract System.Windows.Forms.GridItemCollection get_GridItems();
public abstract System.Windows.Forms.GridItemType get_GridItemType();
public abstract System.String get_Label();
public abstract System.Windows.Forms.GridItem get_Parent();
public abstract System.ComponentModel.PropertyDescriptor get_PropertyDescriptor();
public abstract System.Object get_Value();
```

Public Methods

```
public abstract boolean Select();
```

Protected Constructors

```
protected GridItem();
```

GridItemCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class GridItemCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public System.Windows.Forms.GridItem get_Item(
   int index
);
public System.Windows.Forms.GridItem get_Item(
   System.String label
);
```

Public Methods

```
public final System.Collections.IEnumerator GetEnumerator();
```

Public Fields

```
public static System.Windows.Forms.GridItemCollection Empty;
```

GridItemType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum GridItemType
```

Members

ArrayValue

Category

Property

Root

GridTablesFactory Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class GridTablesFactory
   extends System.Object
```

Public Methods

```
public static System.Windows.Forms.DataGridTableStyle[] CreateGridTables(
   System.Windows.Forms.DataGridTableStyle gridTable,
   System.Object dataSource,
   System.String dataMember,
   System.Windows.Forms.BindingContext bindingManager
);
```

GridTableStylesCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ListBindableAttribute(ListBindable = false) */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class GridTableStylesCollection
   extends System.Windows.Forms.BaseCollection
   implements System.Collections.IList
```

Public Properties

```
public System.Windows.Forms.DataGridTableStyle get_Item(
   int index
);
public System.Windows.Forms.DataGridTableStyle get_Item(
   System.String tableName
);
```

Public Methods

```
public int Add(
   System.Windows.Forms.DataGridTableStyle table
);
public void AddRange(
   System.Windows.Forms.DataGridTableStyle[] tables
);
public void Clear();
public boolean Contains(
   System.String name
);
public boolean Contains(
   System.Windows.Forms.DataGridTableStyle table
);
public void Remove(
   System.Windows.Forms.DataGridTableStyle table
);
public void RemoveAt(
   int index
);
```

Public Events

```
public void add_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
public void remove_CollectionChanged(
   System.ComponentModel.CollectionChangeEventHandler value
);
```

Protected Properties

```
protected System.Collections.ArrayList get_List();
```

Protected Methods

```
protected void OnCollectionChanged(
   System.ComponentModel.CollectionChangeEventArgs ccevent
);
```

GroupBox Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.GroupBoxDesigner,
System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Enter") */
public class GroupBox
   extends System.Windows.Forms.Control
```

Public Constructors

```
public GroupBox();
```

Public Properties

```
public boolean get_AllowDrop();
public void set_AllowDrop(
   boolean value
);
public System.Drawing.Rectangle get_DisplayRectangle();
public System.Windows.Forms.FlatStyle get_FlatStyle();
public void set_FlatStyle(
   System.Windows.Forms.FlatStyle value
);
public boolean get_TabStop();
public void set_TabStop(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Methods

```
public System.String ToString();
```

Public Events

```
public void add_Click(
   System.EventHandler value
);
public void remove_Click(
   System.EventHandler value
);
public void add_DoubleClick(
   System.EventHandler value
);
public void remove_DoubleClick(
   System.EventHandler value
);
public void add_KeyDown(
```

```
   System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyDown(
   System.Windows.Forms.KeyEventHandler value
);
public void add_KeyPress(
   System.Windows.Forms.KeyPressEventHandler value
);
public void remove_KeyPress(
   System.Windows.Forms.KeyPressEventHandler value
);
public void add_KeyUp(
   System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyUp(
   System.Windows.Forms.KeyEventHandler value
);
public void add_MouseDown(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseDown(
   System.Windows.Forms.MouseEventHandler value
);
public void add_MouseEnter(
   System.EventHandler value
);
public void remove_MouseEnter(
   System.EventHandler value
);
public void add_MouseLeave(
   System.EventHandler value
);
public void remove_MouseLeave(
   System.EventHandler value
);
public void add_MouseMove(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseMove(
   System.Windows.Forms.MouseEventHandler value
);
public void add_MouseUp(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseUp(
   System.Windows.Forms.MouseEventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnPaint(
   System.Windows.Forms.PaintEventArgs e
);
protected boolean ProcessMnemonic(
   char charCode
);
```

```
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

Help Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Help
   extends System.Object
```

Public Methods

```
public static void ShowHelp(
   System.Windows.Forms.Control parent,
   System.String url
);
public static void ShowHelp(
   System.Windows.Forms.Control parent,
   System.String url,
   System.String keyword
);
public static void ShowHelp(
   System.Windows.Forms.Control parent,
   System.String url,
   System.Windows.Forms.HelpNavigator navigator
);
public static void ShowHelp(
   System.Windows.Forms.Control parent,
   System.String url,
   System.Windows.Forms.HelpNavigator command,
   System.Object param
);
public static void ShowHelpIndex(
   System.Windows.Forms.Control parent,
   System.String url
);
public static void ShowPopup(
   System.Windows.Forms.Control parent,
   System.String caption,
   System.Drawing.Point location
);
```

HelpEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class HelpEventArgs
   extends System.EventArgs
```

Public Constructors

```
public HelpEventArgs(
   System.Drawing.Point mousePos
);
```

Public Properties

```
public boolean get_Handled();
public void set_Handled(
   boolean value
);
public System.Drawing.Point get_MousePos();
```

HelpEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void HelpEventHandler(
   System.Object sender,
   System.Windows.Forms.HelpEventArgs hlpevent
);
```

HelpNavigator Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum HelpNavigator
```

Members

AssociateIndex

Find

Index

KeywordIndex

TableOfContents

Topic

HelpProvider Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ProvidePropertyAttribute(PropertyName = "HelpKeyword", R
eceiverTypeName = "System.Windows.Forms.Control, System.Windows.Forms, Version=1.0.3300.0, Cu
lture=neutral, PublicKeyToken=b77a5c561934e089") */
/** @attribute System.ComponentModel.ProvidePropertyAttribute(PropertyName = "HelpString", Re
ceiverTypeName = "System.Windows.Forms.Control, System.Windows.Forms, Version=1.0.3300.0, Cul
ture=neutral, PublicKeyToken=b77a5c561934e089") */
/** @attribute System.ComponentModel.ProvidePropertyAttribute(PropertyName = "HelpNavigator",
ReceiverTypeName = "System.Windows.Forms.Control, System.Windows.Forms, Version=1.0.3300.0, C
ulture=neutral, PublicKeyToken=b77a5c561934e089") */
/** @attribute System.ComponentModel.ToolboxItemFilterAttribute(FilterString = "System.Window
s.Forms", FilterType = Allow) */
/** @attribute System.ComponentModel.ProvidePropertyAttribute(PropertyName = "ShowHelp", Rece
iverTypeName = "System.Windows.Forms.Control, System.Windows.Forms, Version=1.0.3300.0, Cultu
re=neutral, PublicKeyToken=b77a5c561934e089") */
public class HelpProvider
   extends System.ComponentModel.Component
   implements System.ComponentModel.IExtenderProvider
```

Public Constructors

```
public HelpProvider();
```

Public Properties

```
public System.String get_HelpNamespace();
public void set_HelpNamespace(
   System.String value
);
```

Public Methods

```
public boolean CanExtend(
   System.Object target
);
/** @attribute System.ComponentModel.DefaultValueAttribute() */
/** @attribute System.Windows.Forms.SRDescriptionAttribute(Description = "Determines the Help
keyword associated with this control.") */
/** @attribute System.ComponentModel.LocalizableAttribute(IsLocalizable = true) */
public System.String GetHelpKeyword(
   System.Windows.Forms.Control ctl
);
/** @attribute System.ComponentModel.DefaultValueAttribute(Value = AssociateIndex) */
/** @attribute System.ComponentModel.LocalizableAttribute(IsLocalizable = true) */
/** @attribute System.Windows.Forms.SRDescriptionAttribute(Description = "Determines the kind
of help associated with this control.") */
public System.Windows.Forms.HelpNavigator GetHelpNavigator(
   System.Windows.Forms.Control ctl
);
/** @attribute System.Windows.Forms.SRDescriptionAttribute(Description = "Determines the help
string associated with this control.") */
/** @attribute System.ComponentModel.DefaultValueAttribute() */
/** @attribute System.ComponentModel.LocalizableAttribute(IsLocalizable = true) */
public System.String GetHelpString(
```

```
   System.Windows.Forms.Control ctl
);
/** @attribute System.Windows.Forms.SRDescriptionAttribute(Description = "Determines if help
should be displayed for this control.") */
/** @attribute System.ComponentModel.LocalizableAttribute(IsLocalizable = true) */
public boolean GetShowHelp(
   System.Windows.Forms.Control ctl
);
public void ResetShowHelp(
   System.Windows.Forms.Control ctl
);
public void SetHelpKeyword(
   System.Windows.Forms.Control ctl,
   System.String keyword
);
public void SetHelpNavigator(
   System.Windows.Forms.Control ctl,
   System.Windows.Forms.HelpNavigator navigator
);
public void SetHelpString(
   System.Windows.Forms.Control ctl,
   System.String helpString
);
public void SetShowHelp(
   System.Windows.Forms.Control ctl,
   boolean value
);
public System.String ToString();
```

HorizontalAlignment Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum HorizontalAlignment
```

Members

Center

Left

Right

HScrollBar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HScrollBar
   extends System.Windows.Forms.ScrollBar
```

Public Constructors

```
public HScrollBar();
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
```

IButtonControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IButtonControl
```

Public Properties

```
System.Windows.Forms.DialogResult get_DialogResult();
void set_DialogResult(
   System.Windows.Forms.DialogResult value
);
```

Public Methods

```
void NotifyDefault(
   boolean value
);
void PerformClick();
```

ICommandExecutor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface ICommandExecutor
```

Public Methods

```
void Execute();
```

IComponentEditorPageSite Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IComponentEditorPageSite
```

Public Methods

```
System.Windows.Forms.Control GetControl();
void SetDirty();
```

IContainerControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IContainerControl
```

Public Properties

```
System.Windows.Forms.Control get_ActiveControl();
void set_ActiveControl(
   System.Windows.Forms.Control value
);
```

Public Methods

```
boolean ActivateControl(
   System.Windows.Forms.Control active
);
```

IDataGridColumnStyleEditingNotificationService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDataGridColumnStyleEditingNotificationService
```

Public Methods

```
void ColumnStartedEditing(
   System.Windows.Forms.Control editingControl
);
```

IDataGridEditingService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IDataGridEditingService
```

Public Methods

```
boolean BeginEdit(
   System.Windows.Forms.DataGridColumnStyle gridColumn,
   int rowNumber
);
boolean EndEdit(
   System.Windows.Forms.DataGridColumnStyle gridColumn,
   int rowNumber,
   boolean shouldAbort
);
```

IDataObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public interface IDataObject
```

Public Methods

```
System.Object GetData(
   System.String format
);
System.Object GetData(
   System.Type format
);
System.Object GetData(
   System.String format,
   boolean autoConvert
);
boolean GetDataPresent(
   System.String format
);
boolean GetDataPresent(
   System.Type format
);
boolean GetDataPresent(
   System.String format,
   boolean autoConvert
);
System.String[] GetFormats();
System.String[] GetFormats(
   boolean autoConvert
);
void SetData(
   System.Object data
);
void SetData(
   System.String format,
   System.Object data
);
void SetData(
   System.Type format,
   System.Object data
);
void SetData(
   System.String format,
   boolean autoConvert,
   System.Object data
);
```

IFeatureSupport Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IFeatureSupport
```

Public Methods

```
System.Version GetVersionPresent(
   System.Object feature
);
boolean IsPresent(
   System.Object feature
);
boolean IsPresent(
   System.Object feature,
   System.Version minimumVersion
);
```

IFileReaderService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IFileReaderService
```

Public Methods

```
System.IO.Stream OpenFileFromSource(
   System.String relativePath
);
```

ImageIndexConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ImageIndexConverter
   extends System.ComponentModel.Int32Converter
```

Public Constructors

```
public ImageIndexConverter();
```

Public Methods

```
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

Protected Properties

```
protected boolean get_IncludeNoneAsStandardValue();
```

ImageList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Images") */
/** @attribute System.ComponentModel.ToolboxItemFilterAttribute(FilterString = "System.Window
s.Forms", FilterType = Allow) */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Windo
ws.Forms.ImageListConverter, System.Windows.Forms, Version=1.0.3300.0, Culture=neutral, Publi
cKeyToken=b77a5c561934e089") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.ImageListDesigner,
System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class ImageList
   extends System.ComponentModel.Component
```

Public Constructors

```
public ImageList();
public ImageList(
   System.ComponentModel.IContainer container
);
```

Public Properties

```
public System.Windows.Forms.ColorDepth get_ColorDepth();
public void set_ColorDepth(
   System.Windows.Forms.ColorDepth value
);
public System.IntPtr get_Handle();
public boolean get_HandleCreated();
public System.Windows.Forms.ImageList+ImageCollection get_Images();
public System.Drawing.Size get_ImageSize();
public void set_ImageSize(
   System.Drawing.Size value
);
public System.Windows.Forms.ImageListStreamer get_ImageStream();
public void set_ImageStream(
   System.Windows.Forms.ImageListStreamer value
);
public System.Drawing.Color get_TransparentColor();
public void set_TransparentColor(
   System.Drawing.Color value
);
```

Public Methods

```
public void Draw(
   System.Drawing.Graphics g,
   System.Drawing.Point pt,
   int index
);
public void Draw(
   System.Drawing.Graphics g,
   int x,
   int y,
   int index
```

```
);
public void Draw(
   System.Drawing.Graphics g,
   int x,
   int y,
   int width,
   int height,
   int index
);
public System.String ToString();
```

Public Events

```
public void add_RecreateHandle(
   System.EventHandler value
);
public void remove_RecreateHandle(
   System.EventHandler value
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

ImageList.ImageCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Windows.Forms.Design.ImageCollectionEditor, System.Desig
n, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ImageCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Properties

```
public final int get_Count();
public boolean get_Empty();
public final boolean get_IsReadOnly();
public System.Drawing.Image get_Item(
   int index
);
public void set_Item(
   int index,
   System.Drawing.Image value
);
```

Public Methods

```
public void Add(
   System.Drawing.Icon value
);
public void Add(
   System.Drawing.Image value
);
public int Add(
   System.Drawing.Image value,
   System.Drawing.Color transparentColor
);
public int AddStrip(
   System.Drawing.Image value
);
public final void Clear();
public boolean Contains(
   System.Drawing.Image image
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Drawing.Image image
);
public void Remove(
   System.Drawing.Image image
);
public final void RemoveAt(
   int index
);
```

ImageListStreamer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class ImageListStreamer
   extends System.Object
   implements System.Runtime.Serialization.ISerializable
```

Public Methods

```
public final void GetObjectData(
   System.Runtime.Serialization.SerializationInfo si,
   System.Runtime.Serialization.StreamingContext context
);
```

ImeMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum ImeMode
```

Members

Alpha

AlphaFull

Disable

Hangul

HangulFull

Hiragana

Inherit

Katakana

KatakanaHalf

NoControl

Off

On

IMessageFilter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IMessageFilter
```

Public Methods

```
boolean PreFilterMessage(
   System.Windows.Forms.Message& m
);
```

InputLanguage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InputLanguage
   extends System.Object
```

Public Properties

```
public System.Globalization.CultureInfo get_Culture();
public static System.Windows.Forms.InputLanguage get_CurrentInputLanguage();
public static void set_CurrentInputLanguage(
   System.Windows.Forms.InputLanguage value
);
public static System.Windows.Forms.InputLanguage get_DefaultInputLanguage();
public System.IntPtr get_Handle();
public static System.Windows.Forms.InputLanguageCollection get_InstalledInputLanguages();
public System.String get_LayoutName();
```

Public Methods

```
public boolean Equals(
   System.Object value
);
public static System.Windows.Forms.InputLanguage FromCulture(
   System.Globalization.CultureInfo culture
);
public int GetHashCode();
```

InputLanguageChangedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InputLanguageChangedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public InputLanguageChangedEventArgs(
   System.Globalization.CultureInfo culture,
   ubyte charSet
);
public InputLanguageChangedEventArgs(
   System.Windows.Forms.InputLanguage inputLanguage,
   ubyte charSet
);
```

Public Properties

```
public ubyte get_CharSet();
public System.Globalization.CultureInfo get_Culture();
public System.Windows.Forms.InputLanguage get_InputLanguage();
```

InputLanguageChangedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void InputLanguageChangedEventHandler(
   System.Object sender,
   System.Windows.Forms.InputLanguageChangedEventArgs e
);
```

InputLanguageChangingEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InputLanguageChangingEventArgs
   extends System.ComponentModel.CancelEventArgs
```

Public Constructors

```
public InputLanguageChangingEventArgs(
   System.Globalization.CultureInfo culture,
   boolean sysCharSet
);
public InputLanguageChangingEventArgs(
   System.Windows.Forms.InputLanguage inputLanguage,
   boolean sysCharSet
);
```

Public Properties

```
public System.Globalization.CultureInfo get_Culture();
public System.Windows.Forms.InputLanguage get_InputLanguage();
public boolean get_SysCharSet();
```

InputLanguageChangingEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void InputLanguageChangingEventHandler(
   System.Object sender,
   System.Windows.Forms.InputLanguageChangingEventArgs e
);
```

InputLanguageCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class InputLanguageCollection
   extends System.Collections.ReadOnlyCollectionBase
```

Public Properties

```
public System.Windows.Forms.InputLanguage get_Item(
   int index
);
```

Public Methods

```
public boolean Contains(
   System.Windows.Forms.InputLanguage value
);
public void CopyTo(
   System.Windows.Forms.InputLanguage[] array,
   int index
);
public int IndexOf(
   System.Windows.Forms.InputLanguage value
);
```

InvalidateEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class InvalidateEventArgs
   extends System.EventArgs
```

Public Constructors

```
public InvalidateEventArgs(
   System.Drawing.Rectangle invalidRect
);
```

Public Properties

```
public System.Drawing.Rectangle get_InvalidRect();
```

InvalidateEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void InvalidateEventHandler(
   System.Object sender,
   System.Windows.Forms.InvalidateEventArgs e
);
```

ItemActivation Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ItemActivation
```

Members

OneClick

Standard

TwoClick

ItemBoundsPortion Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ItemBoundsPortion
```

Members

Entire

Icon

ItemOnly

Label

ItemChangedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ItemChangedEventArgs
   extends System.EventArgs
```

Public Properties

```
public int get_Index();
```

ItemChangedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ItemChangedEventHandler(
   System.Object sender,
   System.Windows.Forms.ItemChangedEventArgs e
);
```

ItemCheckEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class ItemCheckEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ItemCheckEventArgs(
   int index,
   System.Windows.Forms.CheckState newCheckValue,
   System.Windows.Forms.CheckState currentValue
);
```

Public Properties

```
public System.Windows.Forms.CheckState get_CurrentValue();
public int get_Index();
public System.Windows.Forms.CheckState get_NewValue();
public void set_NewValue(
   System.Windows.Forms.CheckState value
);
```

ItemCheckEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ItemCheckEventHandler(
   System.Object sender,
   System.Windows.Forms.ItemCheckEventArgs e
);
```

ItemDragEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class ItemDragEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ItemDragEventArgs(
   System.Windows.Forms.MouseButtons button
);
public ItemDragEventArgs(
   System.Windows.Forms.MouseButtons button,
   System.Object item
);
```

Public Properties

```
public System.Windows.Forms.MouseButtons get_Button();
public System.Object get_Item();
```

ItemDragEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ItemDragEventHandler(
   System.Object sender,
   System.Windows.Forms.ItemDragEventArgs e
);
```

IWin32Window Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "458AB8A2-A1EA-4d7b-8EBE-
DEE5D3D9442C") */
/** @attribute System.Runtime.InteropServices.InterfaceTypeAttribute(Value = InterfaceIsIUnkn
own) */
public interface IWin32Window
```

Public Properties

```
System.IntPtr get_Handle();
```

IWindowTarget Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IWindowTarget
```

Public Methods

```
void OnHandleChange(
   System.IntPtr newHandle
);
void OnMessage(
   System.Windows.Forms.Message& m
);
```

KeyEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class KeyEventArgs
   extends System.EventArgs
```

Public Constructors

```
public KeyEventArgs(
   System.Windows.Forms.Keys keyData
);
```

Public Properties

```
public boolean get_Alt();
public boolean get_Control();
public boolean get_Handled();
public void set_Handled(
   boolean value
);
public System.Windows.Forms.Keys get_KeyCode();
public System.Windows.Forms.Keys get_KeyData();
public int get_KeyValue();
public System.Windows.Forms.Keys get_Modifiers();
public boolean get_Shift();
```

KeyEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void KeyEventHandler(
   System.Object sender,
   System.Windows.Forms.KeyEventArgs e
);
```

KeyPressEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class KeyPressEventArgs
   extends System.EventArgs
```

Public Constructors

```
public KeyPressEventArgs(
   char keyChar
);
```

Public Properties

```
public boolean get_Handled();
public void set_Handled(
   boolean value
);
public char get_KeyChar();
```

KeyPressEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void KeyPressEventHandler(
   System.Object sender,
   System.Windows.Forms.KeyPressEventArgs e
);
```

Keys Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Windo
ws.Forms.KeysConverter, System.Windows.Forms, Version=1.0.3300.0, Culture=neutral, PublicKeyT
oken=b77a5c561934e089") */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.FlagsAttribute() */
public enum Keys
```

Members

A

Add

Alt

Apps

Attn

B

Back

BrowserBack

BrowserFavorites

BrowserForward

BrowserHome

BrowserRefresh

BrowserSearch

BrowserStop

C

Cancel

Capital

CapsLock

Clear

Control

ControlKey

Crsel

D

D0

D1

D2

D3

D4

D5

D6

D7

D8

D9

Decimal

Delete

Divide

Down

E

End

Enter

EraseEof

Escape

Execute

Exsel

F

F1

F10

F11

F12

F13

F14

F15

F16

F17

F18

F19

F2

F20

F21

F22

F23

F24

F3

F4

F5

F6

F7

F8

F9

FinalMode

G

H

HanguelMode

HangulMode

HanjaMode

Help

Home

I

IMEAceept

IMEConvert

IMEModeChange

IMENonconvert

Insert

J

JunjaMode

K

KanaMode

KanjiMode

KeyCode

L

LaunchApplication1

LaunchApplication2

LaunchMail

LButton

LControlKey

Left

LineFeed

LMenu

LShiftKey

LWin

M

MButton

MediaNextTrack

MediaPlayPause

MediaPreviousTrack

MediaStop

Menu

Modifiers

Multiply

N

Next

NoName

None

NumLock

NumPad0

NumPad1

NumPad2

NumPad3

NumPad4

NumPad5

NumPad6

NumPad7

NumPad8

NumPad9

O

Oem8

OemBackslash

OemClear

OemCloseBrackets

Oemcomma

OemMinus

OemOpenBrackets

OemPeriod

OemPipe

Oemplus

OemQuestion

OemQuotes

OemSemicolon

Oemtilde

P

Pa1

PageDown

PageUp

Pause

Play

Print

PrintScreen

Prior

ProcessKey

Q

R

RButton

RControlKey

Return

Right

RMenu

RShiftKey

RWin

S

Scroll

Select

SelectMedia

Separator

Shift

ShiftKey

Snapshot

Space

Subtract

T

Tab

U

Up

V

VolumeDown

VolumeMute

VolumeUp

W

X

XButton1

XButton2

Y

Z

Zoom

KeysConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class KeysConverter
   extends System.ComponentModel.TypeConverter
   implements System.Collections.IComparer
```

Public Constructors

```
public KeysConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public final int Compare(
   System.Object a,
   System.Object b
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.ComponentModel.TypeConverter+StandardValuesCollection GetStandardValues(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesExclusive(
   System.ComponentModel.ITypeDescriptorContext context
);
public boolean GetStandardValuesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

Label Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.LabelDesigner, Sys
tem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
public class Label
   extends System.Windows.Forms.Control
```

Public Constructors

```
public Label();
```

Public Properties

```
public boolean get_AutoSize();
public void set_AutoSize(
   boolean value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Windows.Forms.BorderStyle value
);
public System.Windows.Forms.FlatStyle get_FlatStyle();
public void set_FlatStyle(
   System.Windows.Forms.FlatStyle value
);
public System.Drawing.Image get_Image();
public void set_Image(
   System.Drawing.Image value
);
public System.Drawing.ContentAlignment get_ImageAlign();
public void set_ImageAlign(
   System.Drawing.ContentAlignment value
);
public int get_ImageIndex();
public void set_ImageIndex(
   int value
);
public System.Windows.Forms.ImageList get_ImageList();
public void set_ImageList(
   System.Windows.Forms.ImageList value
);
public System.Windows.Forms.ImeMode get_ImeMode();
public void set_ImeMode(
   System.Windows.Forms.ImeMode value
);
public int get_PreferredHeight();
public int get_PreferredWidth();
public boolean get_TabStop();
public void set_TabStop(
   boolean value
);
```

```
public System.Drawing.ContentAlignment get_TextAlign();
public void set_TextAlign(
   System.Drawing.ContentAlignment value
);
public boolean get_UseMnemonic();
public void set_UseMnemonic(
   boolean value
);
```

Public Methods

```
public System.String ToString();
```

Public Events

```
public void add_AutoSizeChanged(
   System.EventHandler value
);
public void remove_AutoSizeChanged(
   System.EventHandler value
);
public void add_KeyDown(
   System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyDown(
   System.Windows.Forms.KeyEventHandler value
);
public void add_KeyPress(
   System.Windows.Forms.KeyPressEventHandler value
);
public void remove_KeyPress(
   System.Windows.Forms.KeyPressEventHandler value
);
public void add_KeyUp(
   System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyUp(
   System.Windows.Forms.KeyEventHandler value
);
public void add_TextAlignChanged(
   System.EventHandler value
);
public void remove_TextAlignChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Windows.Forms.ImeMode get_DefaultImeMode();
protected System.Drawing.Size get_DefaultSize();
protected boolean get_RenderTransparent();
protected void set_RenderTransparent(
   boolean value
);
```

Protected Methods

```
protected System.Drawing.Rectangle CalcImageRenderBounds(
   System.Drawing.Image image,
```

```
   System.Drawing.Rectangle r,
   System.Drawing.ContentAlignment align
);
protected System.Windows.Forms.AccessibleObject CreateAccessibilityInstance();
protected void Dispose(
   boolean disposing
);
protected void DrawImage(
   System.Drawing.Graphics g,
   System.Drawing.Image image,
   System.Drawing.Rectangle r,
   System.Drawing.ContentAlignment align
);
protected void OnAutoSizeChanged(
   System.EventArgs e
);
protected void OnEnabledChanged(
   System.EventArgs e
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnPaint(
   System.Windows.Forms.PaintEventArgs e
);
protected void OnParentChanged(
   System.EventArgs e
);
protected void OnTextAlignChanged(
   System.EventArgs e
);
protected void OnTextChanged(
   System.EventArgs e
);
protected void OnVisibleChanged(
   System.EventArgs e
);
protected boolean ProcessMnemonic(
   char charCode
);
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

LabelEditEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LabelEditEventArgs
   extends System.EventArgs
```

Public Constructors

```
public LabelEditEventArgs(
   int item
);
public LabelEditEventArgs(
   int item,
   System.String label
);
```

Public Properties

```
public boolean get_CancelEdit();
public void set_CancelEdit(
   boolean value
);
public int get_Item();
public System.String get_Label();
```

LabelEditEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void LabelEditEventHandler(
   System.Object sender,
   System.Windows.Forms.LabelEditEventArgs e
);
```

LayoutEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LayoutEventArgs
   extends System.EventArgs
```

Public Constructors

```
public LayoutEventArgs(
   System.Windows.Forms.Control affectedControl,
   System.String affectedProperty
);
```

Public Properties

```
public System.Windows.Forms.Control get_AffectedControl();
public System.String get_AffectedProperty();
```

LayoutEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void LayoutEventHandler(
   System.Object sender,
   System.Windows.Forms.LayoutEventArgs e
);
```

LeftRightAlignment Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum LeftRightAlignment
```

Members

Left

Right

LinkArea Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Windo
ws.Forms.LinkArea+LinkAreaConverter, System.Windows.Forms, Version=1.0.3300.0, Culture=neutra
l, PublicKeyToken=b77a5c561934e089") */
public class LinkArea
   extends System.ValueType
```

Public Constructors

```
public LinkArea(
   int start,
   int length
);
```

Public Properties

```
public boolean get_IsEmpty();
public int get_Length();
public void set_Length(
   int value
);
public int get_Start();
public void set_Start(
   int value
);
```

Public Methods

```
public boolean Equals(
   System.Object o
);
public int GetHashCode();
```

LinkArea.LinkAreaConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class LinkAreaConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public LinkAreaConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.Object CreateInstance(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Collections.IDictionary propertyValues
);
public boolean GetCreateInstanceSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value,
   System.Attribute[] attributes
);
public boolean GetPropertiesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

LinkBehavior Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum LinkBehavior
```

Members

AlwaysUnderline

HoverUnderline

NeverUnderline

SystemDefault

LinkClickedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class LinkClickedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public LinkClickedEventArgs(
   System.String linkText
);
```

Public Properties

```
public System.String get_LinkText();
```

LinkClickedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void LinkClickedEventHandler(
   System.Object sender,
   System.Windows.Forms.LinkClickedEventArgs e
);
```

LinkLabel Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "LinkClicked") */
public class LinkLabel
   extends System.Windows.Forms.Label
   implements System.Windows.Forms.IButtonControl
```

Public Constructors

```
public LinkLabel();
```

Public Properties

```
public System.Drawing.Color get_ActiveLinkColor();
public void set_ActiveLinkColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_DisabledLinkColor();
public void set_DisabledLinkColor(
   System.Drawing.Color value
);
public System.Windows.Forms.LinkArea get_LinkArea();
public void set_LinkArea(
   System.Windows.Forms.LinkArea value
);
public System.Windows.Forms.LinkBehavior get_LinkBehavior();
public void set_LinkBehavior(
   System.Windows.Forms.LinkBehavior value
);
public System.Drawing.Color get_LinkColor();
public void set_LinkColor(
   System.Drawing.Color value
);
public System.Windows.Forms.LinkLabel+LinkCollection get_Links();
public boolean get_LinkVisited();
public void set_LinkVisited(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.Drawing.Color get_VisitedLinkColor();
public void set_VisitedLinkColor(
   System.Drawing.Color value
);
```

Public Events

```
public void add_LinkClicked(
   System.Windows.Forms.LinkLabelLinkClickedEventHandler value
);
public void remove_LinkClicked(
   System.Windows.Forms.LinkLabelLinkClickedEventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.Cursor get_OverrideCursor();
protected void set_OverrideCursor(
   System.Windows.Forms.Cursor value
);
```

Protected Methods

```
protected System.Windows.Forms.AccessibleObject CreateAccessibilityInstance();
protected void CreateHandle();
protected void OnEnabledChanged(
   System.EventArgs e
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnGotFocus(
   System.EventArgs e
);
protected void OnKeyDown(
   System.Windows.Forms.KeyEventArgs e
);
protected void OnLinkClicked(
   System.Windows.Forms.LinkLabelLinkClickedEventArgs e
);
protected void OnLostFocus(
   System.EventArgs e
);
protected void OnMouseDown(
   System.Windows.Forms.MouseEventArgs e
);
protected void OnMouseLeave(
   System.EventArgs e
);
protected void OnMouseMove(
   System.Windows.Forms.MouseEventArgs e
);
protected void OnMouseUp(
   System.Windows.Forms.MouseEventArgs e
);
protected void OnPaint(
   System.Windows.Forms.PaintEventArgs e
);
protected void OnPaintBackground(
   System.Windows.Forms.PaintEventArgs e
);
protected void OnTextAlignChanged(
   System.EventArgs e
);
protected void OnTextChanged(
   System.EventArgs e
);
protected System.Windows.Forms.LinkLabel+Link PointInLink(
   int x,
   int y
);
protected boolean ProcessDialogKey(
   System.Windows.Forms.Keys keyData
);
protected void Select(
   boolean directed,
   boolean forward
```

```
);
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
protected void WndProc(
   System.Windows.Forms.Message& msg
);
```

LinkLabel.Link Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Link
   extends System.Object
```

Public Properties

```
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public int get_Length();
public void set_Length(
   int value
);
public System.Object get_LinkData();
public void set_LinkData(
   System.Object value
);
public int get_Start();
public void set_Start(
   int value
);
public boolean get_Visited();
public void set_Visited(
   boolean value
);
```

LinkLabel.LinkCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class LinkCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public LinkCollection(
   System.Windows.Forms.LinkLabel owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public System.Windows.Forms.LinkLabel+Link get_Item(
   int index
);
public void set_Item(
   int index,
   System.Windows.Forms.LinkLabel+Link value
);
```

Public Methods

```
public System.Windows.Forms.LinkLabel+Link Add(
   int start,
   int length
);
public System.Windows.Forms.LinkLabel+Link Add(
   int start,
   int length,
   System.Object linkData
);
public void Clear();
public boolean Contains(
   System.Windows.Forms.LinkLabel+Link link
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Windows.Forms.LinkLabel+Link link
);
public void Remove(
   System.Windows.Forms.LinkLabel+Link value
);
public final void RemoveAt(
   int index
);
```

LinkLabelLinkClickedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class LinkLabelLinkClickedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public LinkLabelLinkClickedEventArgs(
   System.Windows.Forms.LinkLabel+Link link
);
```

Public Properties

```
public System.Windows.Forms.LinkLabel+Link get_Link();
```

LinkLabelLinkClickedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void LinkLabelLinkClickedEventHandler(
   System.Object sender,
   System.Windows.Forms.LinkLabelLinkClickedEventArgs e
);
```

LinkState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum LinkState
```

Members

Active

Hover

Normal

Visited

ListBindingConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ListBindingConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public ListBindingConverter();
```

Public Methods

```
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.Object CreateInstance(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Collections.IDictionary propertyValues
);
public boolean GetCreateInstanceSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

ListBox Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "SelectedIndexChanged") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Items") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.ListBoxDesigner, S
ystem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class ListBox
   extends System.Windows.Forms.ListControl
```

Public Constructors

```
public ListBox();
```

Public Properties

```
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Windows.Forms.BorderStyle value
);
public int get_ColumnWidth();
public void set_ColumnWidth(
   int value
);
public System.Windows.Forms.DrawMode get_DrawMode();
public void set_DrawMode(
   System.Windows.Forms.DrawMode value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public int get_HorizontalExtent();
public void set_HorizontalExtent(
   int value
);
public boolean get_HorizontalScrollbar();
public void set_HorizontalScrollbar(
   boolean value
);
public boolean get_IntegralHeight();
public void set_IntegralHeight(
   boolean value
);
public int get_ItemHeight();
public void set_ItemHeight(
   int value
);
public System.Windows.Forms.ListBox+ObjectCollection get_Items();
```

```
public boolean get_MultiColumn();
public void set_MultiColumn(
   boolean value
);
public int get_PreferredHeight();
public System.Windows.Forms.RightToLeft get_RightToLeft();
public void set_RightToLeft(
   System.Windows.Forms.RightToLeft value
);
public boolean get_ScrollAlwaysVisible();
public void set_ScrollAlwaysVisible(
   boolean value
);
public int get_SelectedIndex();
public void set_SelectedIndex(
   int value
);
public System.Windows.Forms.ListBox+SelectedIndexCollection get_SelectedIndices();
public System.Object get_SelectedItem();
public void set_SelectedItem(
   System.Object value
);
public System.Windows.Forms.ListBox+SelectedObjectCollection get_SelectedItems();
public System.Windows.Forms.SelectionMode get_SelectionMode();
public void set_SelectionMode(
   System.Windows.Forms.SelectionMode value
);
public boolean get_Sorted();
public void set_Sorted(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public int get_TopIndex();
public void set_TopIndex(
   int value
);
public boolean get_UseTabStops();
public void set_UseTabStops(
   boolean value
);
```

Public Methods

```
public void BeginUpdate();
public void ClearSelected();
public void EndUpdate();
public int FindString(
   System.String s
);
public int FindString(
   System.String s,
   int startIndex
);
public int FindStringExact(
   System.String s
);
public int FindStringExact(
   System.String s,
   int startIndex
);
public int GetItemHeight(
   int index
);
```

```
public System.Drawing.Rectangle GetItemRectangle(
   int index
);
public boolean GetSelected(
   int index
);
public int IndexFromPoint(
   System.Drawing.Point p
);
public int IndexFromPoint(
   int x,
   int y
);
public void SetSelected(
   int index,
   boolean value
);
public System.String ToString();
```

Public Events

```
public void add_Click(
   System.EventHandler value
);
public void remove_Click(
   System.EventHandler value
);
public void add_DrawItem(
   System.Windows.Forms.DrawItemEventHandler value
);
public void remove_DrawItem(
   System.Windows.Forms.DrawItemEventHandler value
);
public void add_MeasureItem(
   System.Windows.Forms.MeasureItemEventHandler value
);
public void remove_MeasureItem(
   System.Windows.Forms.MeasureItemEventHandler value
);
public void add_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void add_SelectedIndexChanged(
   System.EventHandler value
);
public void remove_SelectedIndexChanged(
   System.EventHandler value
);
```

Public Fields

```
public static int DefaultItemHeight;
public static int NoMatches;
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void AddItemsCore(
   System.Object[] value
);
protected System.Windows.Forms.ListBox+ObjectCollection CreateItemCollection();
protected void OnChangeUICues(
   System.Windows.Forms.UICuesEventArgs e
);
protected void OnDataSourceChanged(
   System.EventArgs e
);
protected void OnDisplayMemberChanged(
   System.EventArgs e
);
protected void OnDrawItem(
   System.Windows.Forms.DrawItemEventArgs e
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnHandleDestroyed(
   System.EventArgs e
);
protected void OnMeasureItem(
   System.Windows.Forms.MeasureItemEventArgs e
);
protected void OnParentChanged(
   System.EventArgs e
);
protected void OnResize(
   System.EventArgs e
);
protected void OnSelectedIndexChanged(
   System.EventArgs e
);
protected void RefreshItem(
   int index
);
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
protected void SetItemCore(
   int index,
   System.Object value
);
protected void SetItemsCore(
   System.Collections.IList value
);
protected void Sort();
protected void WmReflectCommand(
   System.Windows.Forms.Message& m
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

ListBox.ObjectCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ListBindableAttribute(ListBindable = false) */
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ObjectCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public ObjectCollection(
   System.Windows.Forms.ListBox owner
);
public ObjectCollection(
   System.Windows.Forms.ListBox owner,
   System.Windows.Forms.ListBox+ObjectCollection value
);
public ObjectCollection(
   System.Windows.Forms.ListBox owner,
   System.Object[] value
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public System.Object get_Item(
   int index
);
public void set_Item(
   int index,
   System.Object value
);
```

Public Methods

```
public int Add(
   System.Object item
);
public void AddRange(
   System.Windows.Forms.ListBox+ObjectCollection value
);
public void AddRange(
   System.Object[] items
);
public void Clear();
public final boolean Contains(
   System.Object value
);
public void CopyTo(
   System.Object[] dest,
   int arrayIndex
);
public final System.Collections.IEnumerator GetEnumerator();
```

```
public final int IndexOf(
   System.Object value
);
public final void Insert(
   int index,
   System.Object item
);
public final void Remove(
   System.Object value
);
public final void RemoveAt(
   int index
);
```

ListBox.SelectedIndexCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SelectedIndexCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public SelectedIndexCollection(
   System.Windows.Forms.ListBox owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public int get_Item(
   int index
);
```

Public Methods

```
public boolean Contains(
   int selectedIndex
);
public final void CopyTo(
   System.Array dest,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   int selectedIndex
);
```

ListBox.SelectedObjectCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SelectedObjectCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public SelectedObjectCollection(
   System.Windows.Forms.ListBox owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public final System.Object get_Item(
   int index
);
public final void set_Item(
   int index,
   System.Object value
);
```

Public Methods

```
public final boolean Contains(
   System.Object selectedObject
);
public final void CopyTo(
   System.Array dest,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public final int IndexOf(
   System.Object selectedObject
);
```

ListControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class ListControl
   extends System.Windows.Forms.Control
```

Public Properties

```
public System.Object get_DataSource();
public void set_DataSource(
   System.Object value
);
public System.String get_DisplayMember();
public void set_DisplayMember(
   System.String value
);
public abstract int get_SelectedIndex();
public abstract void set_SelectedIndex(
   int value
);
public System.Object get_SelectedValue();
public void set_SelectedValue(
   System.Object value
);
public System.String get_ValueMember();
public void set_ValueMember(
   System.String value
);
```

Public Methods

```
public System.String GetItemText(
   System.Object item
);
```

Public Events

```
public void add_DataSourceChanged(
   System.EventHandler value
);
public void remove_DataSourceChanged(
   System.EventHandler value
);
public void add_DisplayMemberChanged(
   System.EventHandler value
);
public void remove_DisplayMemberChanged(
   System.EventHandler value
);
public void add_SelectedValueChanged(
   System.EventHandler value
);
public void remove_SelectedValueChanged(
   System.EventHandler value
);
public void add_ValueMemberChanged(
```

```
   System.EventHandler value
);
public void remove_ValueMemberChanged(
   System.EventHandler value
);
```

Protected Constructors

```
protected ListControl();
```

Protected Properties

```
protected System.Windows.Forms.CurrencyManager get_DataManager();
```

Protected Methods

```
protected System.Object FilterItemOnProperty(
   System.Object item
);
protected System.Object FilterItemOnProperty(
   System.Object item,
   System.String field
);
protected boolean IsInputKey(
   System.Windows.Forms.Keys keyData
);
protected void OnBindingContextChanged(
   System.EventArgs e
);
protected void OnDataSourceChanged(
   System.EventArgs e
);
protected void OnDisplayMemberChanged(
   System.EventArgs e
);
protected void OnSelectedIndexChanged(
   System.EventArgs e
);
protected void OnSelectedValueChanged(
   System.EventArgs e
);
protected void OnValueMemberChanged(
   System.EventArgs e
);
protected abstract void RefreshItem(
   int index
);
protected void SetItemCore(
   int index,
   System.Object value
);
protected abstract void SetItemsCore(
   System.Collections.IList items
);
```

ListView Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "SelectedIndexChanged") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.ListViewDesigner,
System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Items") */
public class ListView
   extends System.Windows.Forms.Control
```

Public Constructors

```
public ListView();
```

Public Properties

```
public System.Windows.Forms.ItemActivation get_Activation();
public void set_Activation(
   System.Windows.Forms.ItemActivation value
);
public System.Windows.Forms.ListViewAlignment get_Alignment();
public void set_Alignment(
   System.Windows.Forms.ListViewAlignment value
);
public boolean get_AllowColumnReorder();
public void set_AllowColumnReorder(
   boolean value
);
public boolean get_AutoArrange();
public void set_AutoArrange(
   boolean value
);
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Windows.Forms.BorderStyle value
);
public boolean get_CheckBoxes();
public void set_CheckBoxes(
   boolean value
);
public System.Windows.Forms.ListView+CheckedIndexCollection get_CheckedIndices();
public System.Windows.Forms.ListView+CheckedListViewItemCollection get_CheckedItems();
public System.Windows.Forms.ListView+ColumnHeaderCollection get_Columns();
public System.Windows.Forms.ListViewItem get_FocusedItem();
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public boolean get_FullRowSelect();
```

```
public void set_FullRowSelect(
   boolean value
);
public boolean get_GridLines();
public void set_GridLines(
   boolean value
);
public System.Windows.Forms.ColumnHeaderStyle get_HeaderStyle();
public void set_HeaderStyle(
   System.Windows.Forms.ColumnHeaderStyle value
);
public boolean get_HideSelection();
public void set_HideSelection(
   boolean value
);
public boolean get_HoverSelection();
public void set_HoverSelection(
   boolean value
);
public System.Windows.Forms.ListView+ListViewItemCollection get_Items();
public boolean get_LabelEdit();
public void set_LabelEdit(
   boolean value
);
public boolean get_LabelWrap();
public void set_LabelWrap(
   boolean value
);
public System.Windows.Forms.ImageList get_LargeImageList();
public void set_LargeImageList(
   System.Windows.Forms.ImageList value
);
public System.Collections.IComparer get_ListViewItemSorter();
public void set_ListViewItemSorter(
   System.Collections.IComparer value
);
public boolean get_MultiSelect();
public void set_MultiSelect(
   boolean value
);
public boolean get_Scrollable();
public void set_Scrollable(
   boolean value
);
public System.Windows.Forms.ListView+SelectedIndexCollection get_SelectedIndices();
public System.Windows.Forms.ListView+SelectedListViewItemCollection get_SelectedItems();
public System.Windows.Forms.ImageList get_SmallImageList();
public void set_SmallImageList(
   System.Windows.Forms.ImageList value
);
public System.Windows.Forms.SortOrder get_Sorting();
public void set_Sorting(
   System.Windows.Forms.SortOrder value
);
public System.Windows.Forms.ImageList get_StateImageList();
public void set_StateImageList(
   System.Windows.Forms.ImageList value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.Windows.Forms.ListViewItem get_TopItem();
public System.Windows.Forms.View get_View();
public void set_View(
   System.Windows.Forms.View value
);
```

Public Methods

```
public void ArrangeIcons();
public void ArrangeIcons(
   System.Windows.Forms.ListViewAlignment value
);
public void BeginUpdate();
public void Clear();
public void EndUpdate();
public void EnsureVisible(
   int index
);
public System.Windows.Forms.ListViewItem GetItemAt(
   int x,
   int y
);
public System.Drawing.Rectangle GetItemRect(
   int index
);
public System.Drawing.Rectangle GetItemRect(
   int index,
   System.Windows.Forms.ItemBoundsPortion portion
);
public void Sort();
public System.String ToString();
```

Public Events

```
public void add_AfterLabelEdit(
   System.Windows.Forms.LabelEditEventHandler value
);
public void remove_AfterLabelEdit(
   System.Windows.Forms.LabelEditEventHandler value
);
public void add_BeforeLabelEdit(
   System.Windows.Forms.LabelEditEventHandler value
);
public void remove_BeforeLabelEdit(
   System.Windows.Forms.LabelEditEventHandler value
);
public void add_ColumnClick(
   System.Windows.Forms.ColumnClickEventHandler value
);
public void remove_ColumnClick(
   System.Windows.Forms.ColumnClickEventHandler value
);
public void add_ItemActivate(
   System.EventHandler value
);
public void remove_ItemActivate(
   System.EventHandler value
);
public void add_ItemCheck(
   System.Windows.Forms.ItemCheckEventHandler value
);
public void remove_ItemCheck(
   System.Windows.Forms.ItemCheckEventHandler value
);
public void add_ItemDrag(
   System.Windows.Forms.ItemDragEventHandler value
);
public void remove_ItemDrag(
   System.Windows.Forms.ItemDragEventHandler value
);
public void add_Paint(
```

```
   System.Windows.Forms.PaintEventHandler value
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void add_SelectedIndexChanged(
   System.EventHandler value
);
public void remove_SelectedIndexChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void CreateHandle();
protected void Dispose(
   boolean disposing
);
protected boolean IsInputKey(
   System.Windows.Forms.Keys keyData
);
protected void OnAfterLabelEdit(
   System.Windows.Forms.LabelEditEventArgs e
);
protected void OnBeforeLabelEdit(
   System.Windows.Forms.LabelEditEventArgs e
);
protected void OnColumnClick(
   System.Windows.Forms.ColumnClickEventArgs e
);
protected void OnEnabledChanged(
   System.EventArgs e
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnHandleDestroyed(
   System.EventArgs e
);
protected void OnItemActivate(
   System.EventArgs e
);
protected void OnItemCheck(
   System.Windows.Forms.ItemCheckEventArgs ice
);
protected void OnItemDrag(
   System.Windows.Forms.ItemDragEventArgs e
);
protected void OnSelectedIndexChanged(
   System.EventArgs e
);
protected void OnSystemColorsChanged(
   System.EventArgs e
);
protected void RealizeProperties();
protected void UpdateExtendedStyles();
```

```
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

ListView.CheckedIndexCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CheckedIndexCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public CheckedIndexCollection(
   System.Windows.Forms.ListView owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public int get_Item(
   int index
);
```

Public Methods

```
public boolean Contains(
   int checkedIndex
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   int checkedIndex
);
```

ListView.CheckedListViewItemCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class CheckedListViewItemCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public CheckedListViewItemCollection(
   System.Windows.Forms.ListView owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public System.Windows.Forms.ListViewItem get_Item(
   int index
);
```

Public Methods

```
public boolean Contains(
   System.Windows.Forms.ListViewItem item
);
public final void CopyTo(
   System.Array dest,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Windows.Forms.ListViewItem item
);
```

ListView.ColumnHeaderCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ColumnHeaderCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public ColumnHeaderCollection(
   System.Windows.Forms.ListView owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public System.Windows.Forms.ColumnHeader get_Item(
   int index
);
```

Public Methods

```
public int Add(
   System.Windows.Forms.ColumnHeader value
);
public System.Windows.Forms.ColumnHeader Add(
   System.String str,
   int width,
   System.Windows.Forms.HorizontalAlignment textAlign
);
public void AddRange(
   System.Windows.Forms.ColumnHeader[] values
);
public void Clear();
public boolean Contains(
   System.Windows.Forms.ColumnHeader value
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Windows.Forms.ColumnHeader value
);
public void Insert(
   int index,
   System.Windows.Forms.ColumnHeader value
);
public void Insert(
   int index,
   System.String str,
   int width,
   System.Windows.Forms.HorizontalAlignment textAlign
);
public void Remove(
   System.Windows.Forms.ColumnHeader column
```

```
);
public void RemoveAt(
   int index
);
```

ListView.ListViewItemCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ListViewItemCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public ListViewItemCollection(
   System.Windows.Forms.ListView owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public System.Windows.Forms.ListViewItem get_Item(
   int displayIndex
);
public void set_Item(
   int displayIndex,
   System.Windows.Forms.ListViewItem value
);
```

Public Methods

```
public System.Windows.Forms.ListViewItem Add(
   System.String text
);
public System.Windows.Forms.ListViewItem Add(
   System.Windows.Forms.ListViewItem value
);
public System.Windows.Forms.ListViewItem Add(
   System.String text,
   int imageIndex
);
public void AddRange(
   System.Windows.Forms.ListViewItem[] values
);
public void Clear();
public boolean Contains(
   System.Windows.Forms.ListViewItem item
);
public final void CopyTo(
   System.Array dest,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Windows.Forms.ListViewItem item
);
public System.Windows.Forms.ListViewItem Insert(
   int index,
```

```
      System.String text
   );
   public System.Windows.Forms.ListViewItem Insert(
      int index,
      System.Windows.Forms.ListViewItem item
   );
   public System.Windows.Forms.ListViewItem Insert(
      int index,
      System.String text,
      int imageIndex
   );
   public void Remove(
      System.Windows.Forms.ListViewItem item
   );
   public void RemoveAt(
      int index
   );
```

ListView.SelectedIndexCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SelectedIndexCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public SelectedIndexCollection(
   System.Windows.Forms.ListView owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public int get_Item(
   int index
);
```

Public Methods

```
public boolean Contains(
   int selectedIndex
);
public final void CopyTo(
   System.Array dest,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   int selectedIndex
);
```

ListView.SelectedListViewItemCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SelectedListViewItemCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public SelectedListViewItemCollection(
   System.Windows.Forms.ListView owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public System.Windows.Forms.ListViewItem get_Item(
   int index
);
```

Public Methods

```
public final void Clear();
public boolean Contains(
   System.Windows.Forms.ListViewItem item
);
public final void CopyTo(
   System.Array dest,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Windows.Forms.ListViewItem item
);
```

ListViewAlignment Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ListViewAlignment
```

Members

Default

Left

SnapToGrid

Top

ListViewItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Windo
ws.Forms.ListViewItemConverter, System.Windows.Forms, Version=1.0.3300.0, Culture=neutral, Pu
blicKeyToken=b77a5c561934e089") */
public class ListViewItem
   extends System.Object
   implements System.ICloneable, System.Runtime.Serialization.ISerializable
```

Public Constructors

```
public ListViewItem();
public ListViewItem(
   System.String text
);
public ListViewItem(
   System.String[] items
);
public ListViewItem(
   System.Windows.Forms.ListViewItem+ListViewSubItem[] subItems,
   int imageIndex
);
public ListViewItem(
   System.String text,
   int imageIndex
);
public ListViewItem(
   System.String[] items,
   int imageIndex
);
public ListViewItem(
   System.String[] items,
   int imageIndex,
   System.Drawing.Color foreColor,
   System.Drawing.Color backColor,
   System.Drawing.Font font
);
```

Public Properties

```
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Rectangle get_Bounds();
public boolean get_Checked();
public void set_Checked(
   boolean value
);
public boolean get_Focused();
public void set_Focused(
   boolean value
);
```

```
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public int get_ImageIndex();
public void set_ImageIndex(
   int value
);
public System.Windows.Forms.ImageList get_ImageList();
public int get_Index();
public System.Windows.Forms.ListView get_ListView();
public boolean get_Selected();
public void set_Selected(
   boolean value
);
public int get_StateImageIndex();
public void set_StateImageIndex(
   int value
);
public System.Windows.Forms.ListViewItem+ListViewSubItemCollection get_SubItems();
public System.Object get_Tag();
public void set_Tag(
   System.Object value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public boolean get_UseItemStyleForSubItems();
public void set_UseItemStyleForSubItems(
   boolean value
);
```

Public Methods

```
public void BeginEdit();
public System.Object Clone();
public void EnsureVisible();
public System.Drawing.Rectangle GetBounds(
   System.Windows.Forms.ItemBoundsPortion portion
);
public void Remove();
public System.String ToString();
```

Protected Methods

```
protected void Deserialize(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
protected void Serialize(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

ListViewItem.ListViewSubItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Windo
ws.Forms.ListViewSubItemConverter, System.Windows.Forms, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089") */
public class ListViewSubItem
   extends System.Object
```

Public Constructors

```
public ListViewSubItem();
public ListViewSubItem(
   System.Windows.Forms.ListViewItem owner,
   System.String text
);
public ListViewSubItem(
   System.Windows.Forms.ListViewItem owner,
   System.String text,
   System.Drawing.Color foreColor,
   System.Drawing.Color backColor,
   System.Drawing.Font font
);
```

Public Properties

```
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Methods

```
public void ResetStyle();
public System.String ToString();
```

ListViewItem.ListViewSubItemCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ListViewSubItemCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public ListViewSubItemCollection(
   System.Windows.Forms.ListViewItem owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public System.Windows.Forms.ListViewItem+ListViewSubItem get_Item(
   int index
);
public void set_Item(
   int index,
   System.Windows.Forms.ListViewItem+ListViewSubItem value
);
```

Public Methods

```
public System.Windows.Forms.ListViewItem+ListViewSubItem Add(
   System.Windows.Forms.ListViewItem+ListViewSubItem item
);
public System.Windows.Forms.ListViewItem+ListViewSubItem Add(
   System.String text
);
public System.Windows.Forms.ListViewItem+ListViewSubItem Add(
   System.String text,
   System.Drawing.Color foreColor,
   System.Drawing.Color backColor,
   System.Drawing.Font font
);
public void AddRange(
   System.Windows.Forms.ListViewItem+ListViewSubItem[] items
);
public void AddRange(
   System.String[] items
);
public void AddRange(
   System.String[] items,
   System.Drawing.Color foreColor,
   System.Drawing.Color backColor,
   System.Drawing.Font font
);
```

```
public final void Clear();
public boolean Contains(
   System.Windows.Forms.ListViewItem+ListViewSubItem subItem
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Windows.Forms.ListViewItem+ListViewSubItem subItem
);
public void Insert(
   int index,
   System.Windows.Forms.ListViewItem+ListViewSubItem item
);
public void Remove(
   System.Windows.Forms.ListViewItem+ListViewSubItem item
);
public final void RemoveAt(
   int index
);
```

ListViewItemConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ListViewItemConverter
   extends System.ComponentModel.ExpandableObjectConverter
```

Public Constructors

```
public ListViewItemConverter();
```

Public Methods

```
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

MainMenu Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemFilterAttribute(FilterString = "System.Window
s.Forms.MainMenu", FilterType = Allow) */
public class MainMenu
   extends System.Windows.Forms.Menu
```

Public Constructors

```
public MainMenu();
public MainMenu(
   System.Windows.Forms.MenuItem[] items
);
```

Public Properties

```
public System.Windows.Forms.RightToLeft get_RightToLeft();
public void set_RightToLeft(
   System.Windows.Forms.RightToLeft value
);
```

Public Methods

```
public System.Windows.Forms.MainMenu CloneMenu();
public System.Windows.Forms.Form GetForm();
public System.String ToString();
```

Protected Methods

```
protected System.IntPtr CreateMenuHandle();
protected void Dispose(
   boolean disposing
);
```

MdiClient Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
public class MdiClient
   extends System.Windows.Forms.Control
```

Public Constructors

```
public MdiClient();
```

Public Properties

```
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.Form[] get_MdiChildren();
```

Public Methods

```
public void LayoutMdi(
   System.Windows.Forms.MdiLayout value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
```

Protected Methods

```
protected System.Windows.Forms.Control+ControlCollection CreateControlsInstance();
protected void OnResize(
   System.EventArgs e
);
protected void ScaleCore(
   float dx,
   float dy
);
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

MdiClient.ControlCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ControlCollection
   extends System.Windows.Forms.Control+ControlCollection
```

Public Constructors

```
public ControlCollection(
   System.Windows.Forms.MdiClient owner
);
```

Public Methods

```
public void Add(
   System.Windows.Forms.Control value
);
public void Remove(
   System.Windows.Forms.Control value
);
```

MdiLayout Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MdiLayout
```

Members

ArrangeIcons

Cascade

TileHorizontal

TileVertical

MeasureItemEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MeasureItemEventArgs
   extends System.EventArgs
```

Public Constructors

```
public MeasureItemEventArgs(
   System.Drawing.Graphics graphics,
   int index
);
public MeasureItemEventArgs(
   System.Drawing.Graphics graphics,
   int index,
   int itemHeight
);
```

Public Properties

```
public System.Drawing.Graphics get_Graphics();
public int get_Index();
public int get_ItemHeight();
public void set_ItemHeight(
   int value
);
public int get_ItemWidth();
public void set_ItemWidth(
   int value
);
```

MeasureItemEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void MeasureItemEventHandler(
   System.Object sender,
   System.Windows.Forms.MeasureItemEventArgs e
);
```

Menu Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "Microsoft.VisualStudio.Windows.Forms.MenuDesig
ner, Microsoft.VisualStudio, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d5
0a3a") */
/** @attribute System.ComponentModel.ToolboxItemFilterAttribute(FilterString = "System.Window
s.Forms", FilterType = Allow) */
/** @attribute System.ComponentModel.ListBindableAttribute(ListBindable = false) */
public abstract class Menu
   extends System.ComponentModel.Component
```

Public Properties

```
public System.IntPtr get_Handle();
public boolean get_IsParent();
public System.Windows.Forms.MenuItem get_MdiListItem();
public System.Windows.Forms.Menu+MenuItemCollection get_MenuItems();
```

Public Methods

```
public System.Windows.Forms.MenuItem FindMenuItem(
   int type,
   System.IntPtr value
);
public System.Windows.Forms.ContextMenu GetContextMenu();
public System.Windows.Forms.MainMenu GetMainMenu();
public void MergeMenu(
   System.Windows.Forms.Menu menuSrc
);
public System.String ToString();
```

Public Fields

```
public static int FindHandle;
public static int FindShortcut;
```

Protected Constructors

```
protected Menu(
   System.Windows.Forms.MenuItem[] items
);
```

Protected Methods

```
protected void CloneMenu(
   System.Windows.Forms.Menu menuSrc
);
protected System.IntPtr CreateMenuHandle();
protected void Dispose(
   boolean disposing
```

```
);
protected int FindMergePosition(
   int mergeOrder
);
protected boolean ProcessCmdKey(
   System.Windows.Forms.Message& msg,
   System.Windows.Forms.Keys keyData
);
```

Menu.MenuItemCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.ListBindableAttribute(ListBindable = false) */
public class MenuItemCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public MenuItemCollection(
   System.Windows.Forms.Menu owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public System.Windows.Forms.MenuItem get_Item(
   int index
);
```

Public Methods

```
public int Add(
   System.Windows.Forms.MenuItem item
);
public System.Windows.Forms.MenuItem Add(
   System.String caption
);
public int Add(
   int index,
   System.Windows.Forms.MenuItem item
);
public System.Windows.Forms.MenuItem Add(
   System.String caption,
   System.EventHandler onClick
);
public System.Windows.Forms.MenuItem Add(
   System.String caption,
   System.Windows.Forms.MenuItem[] items
);
public void AddRange(
   System.Windows.Forms.MenuItem[] items
);
public void Clear();
public boolean Contains(
   System.Windows.Forms.MenuItem value
);
public final void CopyTo(
   System.Array dest,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
```

```
public int IndexOf(
   System.Windows.Forms.MenuItem value
);
public void Remove(
   System.Windows.Forms.MenuItem item
);
public void RemoveAt(
   int index
);
```

MenuGlyph Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MenuGlyph
```

Members

Arrow

Bullet

Checkmark

Max

Min

MenuItem Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Click") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class MenuItem
   extends System.Windows.Forms.Menu
```

Public Constructors

```
public MenuItem();
public MenuItem(
   System.String text
);
public MenuItem(
   System.String text,
   System.EventHandler onClick
);
public MenuItem(
   System.String text,
   System.Windows.Forms.MenuItem[] items
);
public MenuItem(
   System.String text,
   System.EventHandler onClick,
   System.Windows.Forms.Shortcut shortcut
);
public MenuItem(
   System.Windows.Forms.MenuMerge mergeType,
   int mergeOrder,
   System.Windows.Forms.Shortcut shortcut,
   System.String text,
   System.EventHandler onClick,
   System.EventHandler onPopup,
   System.EventHandler onSelect,
   System.Windows.Forms.MenuItem[] items
);
```

Public Properties

```
public boolean get_BarBreak();
public void set_BarBreak(
   boolean value
);
public boolean get_Break();
public void set_Break(
   boolean value
);
public boolean get_Checked();
public void set_Checked(
   boolean value
);
public boolean get_DefaultItem();
public void set_DefaultItem(
   boolean value
);
```

```
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public int get_Index();
public void set_Index(
   int value
);
public boolean get_IsParent();
public boolean get_MdiList();
public void set_MdiList(
   boolean value
);
public int get_MergeOrder();
public void set_MergeOrder(
   int value
);
public System.Windows.Forms.MenuMerge get_MergeType();
public void set_MergeType(
   System.Windows.Forms.MenuMerge value
);
public char get_Mnemonic();
public boolean get_OwnerDraw();
public void set_OwnerDraw(
   boolean value
);
public System.Windows.Forms.Menu get_Parent();
public boolean get_RadioCheck();
public void set_RadioCheck(
   boolean value
);
public System.Windows.Forms.Shortcut get_Shortcut();
public void set_Shortcut(
   System.Windows.Forms.Shortcut value
);
public boolean get_ShowShortcut();
public void set_ShowShortcut(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public boolean get_Visible();
public void set_Visible(
   boolean value
);
```

Public Methods

```
public System.Windows.Forms.MenuItem CloneMenu();
public System.Windows.Forms.MenuItem MergeMenu();
public void MergeMenu(
   System.Windows.Forms.MenuItem itemSrc
);
public void PerformClick();
public void PerformSelect();
public System.String ToString();
```

Public Events

```
public void add_Click(
   System.EventHandler value
);
```

```
public void remove_Click(
   System.EventHandler value
);
public void add_DrawItem(
   System.Windows.Forms.DrawItemEventHandler value
);
public void remove_DrawItem(
   System.Windows.Forms.DrawItemEventHandler value
);
public void add_MeasureItem(
   System.Windows.Forms.MeasureItemEventHandler value
);
public void remove_MeasureItem(
   System.Windows.Forms.MeasureItemEventHandler value
);
public void add_Popup(
   System.EventHandler value
);
public void remove_Popup(
   System.EventHandler value
);
public void add_Select(
   System.EventHandler value
);
public void remove_Select(
   System.EventHandler value
);
```

Protected Properties

```
protected int get_MenuID();
```

Protected Methods

```
protected void CloneMenu(
   System.Windows.Forms.MenuItem itemSrc
);
protected void Dispose(
   boolean disposing
);
protected void OnClick(
   System.EventArgs e
);
protected void OnDrawItem(
   System.Windows.Forms.DrawItemEventArgs e
);
protected void OnInitMenuPopup(
   System.EventArgs e
);
protected void OnMeasureItem(
   System.Windows.Forms.MeasureItemEventArgs e
);
protected void OnPopup(
   System.EventArgs e
);
protected void OnSelect(
   System.EventArgs e
);
```

MenuMerge Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MenuMerge
```

Members

Add

MergeItems

Remove

Replace

Message Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Message
   extends System.ValueType
```

Public Properties

```
public System.IntPtr get_HWnd();
public void set_HWnd(
   System.IntPtr value
);
public System.IntPtr get_LParam();
public void set_LParam(
   System.IntPtr value
);
public int get_Msg();
public void set_Msg(
   int value
);
public System.IntPtr get_Result();
public void set_Result(
   System.IntPtr value
);
public System.IntPtr get_WParam();
public void set_WParam(
   System.IntPtr value
);
```

Public Methods

```
public static System.Windows.Forms.Message Create(
   System.IntPtr hWnd,
   int msg,
   System.IntPtr wparam,
   System.IntPtr lparam
);
public boolean Equals(
   System.Object o
);
public int GetHashCode();
public System.Object GetLParam(
   System.Type cls
);
public System.String ToString();
```

MessageBox Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MessageBox
   extends System.Object
```

Public Methods

```
public static System.Windows.Forms.DialogResult Show(
   System.String text
);
public static System.Windows.Forms.DialogResult Show(
   System.String text,
   System.String caption
);
public static System.Windows.Forms.DialogResult Show(
   System.Windows.Forms.IWin32Window owner,
   System.String text
);
public static System.Windows.Forms.DialogResult Show(
   System.String text,
   System.String caption,
   System.Windows.Forms.MessageBoxButtons buttons
);
public static System.Windows.Forms.DialogResult Show(
   System.Windows.Forms.IWin32Window owner,
   System.String text,
   System.String caption
);
public static System.Windows.Forms.DialogResult Show(
   System.String text,
   System.String caption,
   System.Windows.Forms.MessageBoxButtons buttons,
   System.Windows.Forms.MessageBoxIcon icon
);
public static System.Windows.Forms.DialogResult Show(
   System.Windows.Forms.IWin32Window owner,
   System.String text,
   System.String caption,
   System.Windows.Forms.MessageBoxButtons buttons
);
public static System.Windows.Forms.DialogResult Show(
   System.String text,
   System.String caption,
   System.Windows.Forms.MessageBoxButtons buttons,
   System.Windows.Forms.MessageBoxIcon icon,
   System.Windows.Forms.MessageBoxDefaultButton defaultButton
);
public static System.Windows.Forms.DialogResult Show(
   System.Windows.Forms.IWin32Window owner,
   System.String text,
   System.String caption,
   System.Windows.Forms.MessageBoxButtons buttons,
   System.Windows.Forms.MessageBoxIcon icon
);
public static System.Windows.Forms.DialogResult Show(
   System.String text,
   System.String caption,
   System.Windows.Forms.MessageBoxButtons buttons,
   System.Windows.Forms.MessageBoxIcon icon,
```

```
   System.Windows.Forms.MessageBoxDefaultButton defaultButton,
   System.Windows.Forms.MessageBoxOptions options
);
public static System.Windows.Forms.DialogResult Show(
   System.Windows.Forms.IWin32Window owner,
   System.String text,
   System.String caption,
   System.Windows.Forms.MessageBoxButtons buttons,
   System.Windows.Forms.MessageBoxIcon icon,
   System.Windows.Forms.MessageBoxDefaultButton defaultButton
);
public static System.Windows.Forms.DialogResult Show(
   System.Windows.Forms.IWin32Window owner,
   System.String text,
   System.String caption,
   System.Windows.Forms.MessageBoxButtons buttons,
   System.Windows.Forms.MessageBoxIcon icon,
   System.Windows.Forms.MessageBoxDefaultButton defaultButton,
   System.Windows.Forms.MessageBoxOptions options
);
```

MessageBoxButtons Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MessageBoxButtons
```

Members

AbortRetryIgnore

OK

OKCancel

RetryCancel

YesNo

YesNoCancel

MessageBoxDefaultButton Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MessageBoxDefaultButton
```

Members

Button1

Button2

Button3

MessageBoxIcon Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum MessageBoxIcon
```

Members

Asterisk

Error

Exclamation

Hand

Information

None

Question

Stop

Warning

MessageBoxOptions Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum MessageBoxOptions
```

Members

DefaultDesktopOnly

RightAlign

RtlReading

ServiceNotification

MethodInvoker Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void MethodInvoker();
```

MonthCalendar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "DateChanged") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.MonthCalendarDesig
ner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "SelectionRange") */
public class MonthCalendar
   extends System.Windows.Forms.Control
```

Public Constructors

```
public MonthCalendar();
```

Public Properties

```
public System.DateTime[] get_AnnuallyBoldedDates();
public void set_AnnuallyBoldedDates(
   System.DateTime[] value
);
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.DateTime[] get_BoldedDates();
public void set_BoldedDates(
   System.DateTime[] value
);
public System.Drawing.Size get_CalendarDimensions();
public void set_CalendarDimensions(
   System.Drawing.Size value
);
public System.Windows.Forms.Day get_FirstDayOfWeek();
public void set_FirstDayOfWeek(
   System.Windows.Forms.Day value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.Windows.Forms.ImeMode get_ImeMode();
public void set_ImeMode(
   System.Windows.Forms.ImeMode value
);
public System.DateTime get_MaxDate();
public void set_MaxDate(
   System.DateTime value
);
public int get_MaxSelectionCount();
public void set_MaxSelectionCount(
   int value
);
public System.DateTime get_MinDate();
```

```
public void set_MinDate(
   System.DateTime value
);
public System.DateTime[] get_MonthlyBoldedDates();
public void set_MonthlyBoldedDates(
   System.DateTime[] value
);
public int get_ScrollChange();
public void set_ScrollChange(
   int value
);
public System.DateTime get_SelectionEnd();
public void set_SelectionEnd(
   System.DateTime value
);
public System.Windows.Forms.SelectionRange get_SelectionRange();
public void set_SelectionRange(
   System.Windows.Forms.SelectionRange value
);
public System.DateTime get_SelectionStart();
public void set_SelectionStart(
   System.DateTime value
);
public boolean get_ShowToday();
public void set_ShowToday(
   boolean value
);
public boolean get_ShowTodayCircle();
public void set_ShowTodayCircle(
   boolean value
);
public boolean get_ShowWeekNumbers();
public void set_ShowWeekNumbers(
   boolean value
);
public System.Drawing.Size get_SingleMonthSize();
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.Drawing.Color get_TitleBackColor();
public void set_TitleBackColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_TitleForeColor();
public void set_TitleForeColor(
   System.Drawing.Color value
);
public System.DateTime get_TodayDate();
public void set_TodayDate(
   System.DateTime value
);
public boolean get_TodayDateSet();
public System.Drawing.Color get_TrailingForeColor();
public void set_TrailingForeColor(
   System.Drawing.Color value
);
```

Public Methods

```
public void AddAnnuallyBoldedDate(
   System.DateTime date
);
public void AddBoldedDate(
   System.DateTime date
);
```

```
public void AddMonthlyBoldedDate(
   System.DateTime date
);
public System.Windows.Forms.SelectionRange GetDisplayRange(
   boolean visible
);
public System.Windows.Forms.MonthCalendar+HitTestInfo HitTest(
   System.Drawing.Point point
);
public System.Windows.Forms.MonthCalendar+HitTestInfo HitTest(
   int x,
   int y
);
public void RemoveAllAnnuallyBoldedDates();
public void RemoveAllBoldedDates();
public void RemoveAllMonthlyBoldedDates();
public void RemoveAnnuallyBoldedDate(
   System.DateTime date
);
public void RemoveBoldedDate(
   System.DateTime date
);
public void RemoveMonthlyBoldedDate(
   System.DateTime date
);
public void SetCalendarDimensions(
   int x,
   int y
);
public void SetDate(
   System.DateTime date
);
public void SetSelectionRange(
   System.DateTime date1,
   System.DateTime date2
);
public System.String ToString();
public void UpdateBoldedDates();
```

Public Events

```
public void add_Click(
   System.EventHandler value
);
public void remove_Click(
   System.EventHandler value
);
public void add_DateChanged(
   System.Windows.Forms.DateRangeEventHandler value
);
public void remove_DateChanged(
   System.Windows.Forms.DateRangeEventHandler value
);
public void add_DateSelected(
   System.Windows.Forms.DateRangeEventHandler value
);
public void remove_DateSelected(
   System.Windows.Forms.DateRangeEventHandler value
);
public void add_DoubleClick(
   System.EventHandler value
);
public void remove_DoubleClick(
   System.EventHandler value
);
public void add_Paint(
```

```
   System.Windows.Forms.PaintEventHandler value
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Windows.Forms.ImeMode get_DefaultImeMode();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void CreateHandle();
protected void Dispose(
   boolean disposing
);
protected boolean IsInputKey(
   System.Windows.Forms.Keys keyData
);
protected void OnBackColorChanged(
   System.EventArgs e
);
protected void OnDateChanged(
   System.Windows.Forms.DateRangeEventArgs drevent
);
protected void OnDateSelected(
   System.Windows.Forms.DateRangeEventArgs drevent
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnForeColorChanged(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

MonthCalendar.HitArea Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum HitArea
```

Members

CalendarBackground

Date

DayOfWeek

NextMonthButton

NextMonthDate

Nowhere

PrevMonthButton

PrevMonthDate

TitleBackground

TitleMonth

TitleYear

TodayLink

WeekNumbers

MonthCalendar.HitTestInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class HitTestInfo
   extends System.Object
```

Public Properties

```
public System.Windows.Forms.MonthCalendar+HitArea get_HitArea();
public System.Drawing.Point get_Point();
public System.DateTime get_Time();
```

MouseButtons Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
/** @attribute System.FlagsAttribute() */
public enum MouseButtons
```

Members

Left

Middle

None

Right

XButton1

XButton2

MouseEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class MouseEventArgs
   extends System.EventArgs
```

Public Constructors

```
public MouseEventArgs(
   System.Windows.Forms.MouseButtons button,
   int clicks,
   int x,
   int y,
   int delta
);
```

Public Properties

```
public System.Windows.Forms.MouseButtons get_Button();
public int get_Clicks();
public int get_Delta();
public int get_X();
public int get_Y();
```

MouseEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void MouseEventHandler(
   System.Object sender,
   System.Windows.Forms.MouseEventArgs e
);
```

NativeWindow Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class NativeWindow
   extends System.MarshalByRefObject
```

Public Constructors

```
public NativeWindow();
```

Public Properties

```
public System.IntPtr get_Handle();
```

Public Methods

```
public void AssignHandle(
   System.IntPtr handle
);
public void CreateHandle(
   System.Windows.Forms.CreateParams cp
);
public void DefWndProc(
   System.Windows.Forms.Message& m
);
public void DestroyHandle();
public static System.Windows.Forms.NativeWindow FromHandle(
   System.IntPtr handle
);
public void ReleaseHandle();
```

Protected Methods

```
protected void Finalize();
protected void OnHandleChange();
protected void OnThreadException(
   System.Exception e
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

NavigateEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class NavigateEventArgs
   extends System.EventArgs
```

Public Constructors

```
public NavigateEventArgs(
   boolean isForward
);
```

Public Properties

```
public boolean get_Forward();
```

NavigateEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void NavigateEventHandler(
   System.Object sender,
   System.Windows.Forms.NavigateEventArgs ne
);
```

NodeLabelEditEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class NodeLabelEditEventArgs
   extends System.EventArgs
```

Public Constructors

```
public NodeLabelEditEventArgs(
   System.Windows.Forms.TreeNode node
);
public NodeLabelEditEventArgs(
   System.Windows.Forms.TreeNode node,
   System.String label
);
```

Public Properties

```
public boolean get_CancelEdit();
public void set_CancelEdit(
   boolean value
);
public System.String get_Label();
public System.Windows.Forms.TreeNode get_Node();
```

NodeLabelEditEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void NodeLabelEditEventHandler(
   System.Object sender,
   System.Windows.Forms.NodeLabelEditEventArgs e
);
```

NotifyIcon Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "MouseDown") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.NotifyIconDesigner
, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.ToolboxItemFilterAttribute(FilterString = "System.Window
s.Forms", FilterType = Allow) */
public class NotifyIcon
   extends System.ComponentModel.Component
```

Public Constructors

```
public NotifyIcon();
public NotifyIcon(
   System.ComponentModel.IContainer container
);
```

Public Properties

```
public System.Windows.Forms.ContextMenu get_ContextMenu();
public void set_ContextMenu(
   System.Windows.Forms.ContextMenu value
);
public System.Drawing.Icon get_Icon();
public void set_Icon(
   System.Drawing.Icon value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public boolean get_Visible();
public void set_Visible(
   boolean value
);
```

Public Events

```
public void add_Click(
   System.EventHandler value
);
public void remove_Click(
   System.EventHandler value
);
public void add_DoubleClick(
   System.EventHandler value
);
public void remove_DoubleClick(
   System.EventHandler value
);
public void add_MouseDown(
   System.Windows.Forms.MouseEventHandler value
);
```

```
public void remove_MouseDown(
   System.Windows.Forms.MouseEventHandler value
);
public void add_MouseMove(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseMove(
   System.Windows.Forms.MouseEventHandler value
);
public void add_MouseUp(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseUp(
   System.Windows.Forms.MouseEventHandler value
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

NumericUpDown Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ValueChanged") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Value") */
public class NumericUpDown
   extends System.Windows.Forms.UpDownBase
   implements System.ComponentModel.ISupportInitialize
```

Public Constructors

```
public NumericUpDown();
```

Public Properties

```
public int get_DecimalPlaces();
public void set_DecimalPlaces(
   int value
);
public boolean get_Hexadecimal();
public void set_Hexadecimal(
   boolean value
);
public System.Decimal get_Increment();
public void set_Increment(
   System.Decimal value
);
public System.Decimal get_Maximum();
public void set_Maximum(
   System.Decimal value
);
public System.Decimal get_Minimum();
public void set_Minimum(
   System.Decimal value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public boolean get_ThousandsSeparator();
public void set_ThousandsSeparator(
   boolean value
);
public System.Decimal get_Value();
public void set_Value(
   System.Decimal value
);
```

Public Methods

```
public final void BeginInit();
public void DownButton();
public final void EndInit();
public System.String ToString();
public void UpButton();
```

Public Events

```
public void add_ValueChanged(
   System.EventHandler value
);
public void remove_ValueChanged(
   System.EventHandler value
);
```

Protected Methods

```
protected System.Windows.Forms.AccessibleObject CreateAccessibilityInstance();
protected void OnTextBoxKeyPress(
   System.Object source,
   System.Windows.Forms.KeyPressEventArgs e
);
protected void OnValueChanged(
   System.EventArgs e
);
protected void ParseEditText();
protected void UpdateEditText();
protected void ValidateEditText();
```

OpacityConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class OpacityConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public OpacityConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

OpenFileDialog Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.OpenFileDialogDesi
gner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") *
/
public class OpenFileDialog
   extends System.Windows.Forms.FileDialog
```

Public Constructors

```
public OpenFileDialog();
```

Public Properties

```
public boolean get_CheckFileExists();
public void set_CheckFileExists(
   boolean value
);
public boolean get_Multiselect();
public void set_Multiselect(
   boolean value
);
public boolean get_ReadOnlyChecked();
public void set_ReadOnlyChecked(
   boolean value
);
public boolean get_ShowReadOnly();
public void set_ShowReadOnly(
   boolean value
);
```

Public Methods

```
public System.IO.Stream OpenFile();
public void Reset();
```

Orientation Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum Orientation
```

Members

Horizontal

Vertical

OSFeature Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class OSFeature
   extends System.Windows.Forms.FeatureSupport
```

Public Properties

```
public static System.Windows.Forms.OSFeature get_Feature();
```

Public Methods

```
public System.Version GetVersionPresent(
   System.Object feature
);
```

Public Fields

```
public static final System.Object LayeredWindows;
public static final System.Object Themes;
```

Protected Constructors

```
protected OSFeature();
```

OwnerDrawPropertyBag Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class OwnerDrawPropertyBag
   extends System.MarshalByRefObject
   implements System.Runtime.Serialization.ISerializable
```

Public Properties

```
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
```

Public Methods

```
public static System.Windows.Forms.OwnerDrawPropertyBag Copy(
   System.Windows.Forms.OwnerDrawPropertyBag value
);
public boolean IsEmpty();
```

PageSetupDialog Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Document") */
public class PageSetupDialog
   extends System.Windows.Forms.CommonDialog
```

Public Constructors

```
public PageSetupDialog();
```

Public Properties

```
public boolean get_AllowMargins();
public void set_AllowMargins(
   boolean value
);
public boolean get_AllowOrientation();
public void set_AllowOrientation(
   boolean value
);
public boolean get_AllowPaper();
public void set_AllowPaper(
   boolean value
);
public boolean get_AllowPrinter();
public void set_AllowPrinter(
   boolean value
);
public System.Drawing.Printing.PrintDocument get_Document();
public void set_Document(
   System.Drawing.Printing.PrintDocument value
);
public System.Drawing.Printing.Margins get_MinMargins();
public void set_MinMargins(
   System.Drawing.Printing.Margins value
);
public System.Drawing.Printing.PageSettings get_PageSettings();
public void set_PageSettings(
   System.Drawing.Printing.PageSettings value
);
public System.Drawing.Printing.PrinterSettings get_PrinterSettings();
public void set_PrinterSettings(
   System.Drawing.Printing.PrinterSettings value
);
public boolean get_ShowHelp();
public void set_ShowHelp(
   boolean value
);
public boolean get_ShowNetwork();
public void set_ShowNetwork(
   boolean value
);
```

Public Methods

```
public void Reset();
```

Protected Methods

```
protected boolean RunDialog(
   System.IntPtr hwndOwner
);
```

PaintEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PaintEventArgs
   extends System.EventArgs
   implements System.IDisposable
```

Public Constructors

```
public PaintEventArgs(
   System.Drawing.Graphics graphics,
   System.Drawing.Rectangle clipRect
);
```

Public Properties

```
public System.Drawing.Rectangle get_ClipRectangle();
public System.Drawing.Graphics get_Graphics();
```

Public Methods

```
public final void Dispose();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

PaintEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void PaintEventHandler(
   System.Object sender,
   System.Windows.Forms.PaintEventArgs e
);
```

Panel Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Paint") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.PanelDesigner, Sys
tem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "BorderStyle") */
public class Panel
   extends System.Windows.Forms.ScrollableControl
```

Public Constructors

```
public Panel();
```

Public Properties

```
public System.Windows.Forms.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Windows.Forms.BorderStyle value
);
public boolean get_TabStop();
public void set_TabStop(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Methods

```
public System.String ToString();
```

Public Events

```
public void add_KeyDown(
   System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyDown(
   System.Windows.Forms.KeyEventHandler value
);
public void add_KeyPress(
   System.Windows.Forms.KeyPressEventHandler value
);
public void remove_KeyPress(
   System.Windows.Forms.KeyPressEventHandler value
);
public void add_KeyUp(
   System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyUp(
   System.Windows.Forms.KeyEventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void OnResize(
   System.EventArgs eventargs
);
```

PictureBox Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Image") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.PictureBoxDesigner
, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class PictureBox
   extends System.Windows.Forms.Control
```

Public Constructors

```
public PictureBox();
```

Public Properties

```
public boolean get_AllowDrop();
public void set_AllowDrop(
   boolean value
);
public System.Windows.Forms.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Windows.Forms.BorderStyle value
);
public boolean get_CausesValidation();
public void set_CausesValidation(
   boolean value
);
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_Image();
public void set_Image(
   System.Drawing.Image value
);
public System.Windows.Forms.ImeMode get_ImeMode();
public void set_ImeMode(
   System.Windows.Forms.ImeMode value
);
public System.Windows.Forms.RightToLeft get_RightToLeft();
public void set_RightToLeft(
   System.Windows.Forms.RightToLeft value
);
public System.Windows.Forms.PictureBoxSizeMode get_SizeMode();
public void set_SizeMode(
   System.Windows.Forms.PictureBoxSizeMode value
);
public int get_TabIndex();
public void set_TabIndex(
   int value
);
public boolean get_TabStop();
public void set_TabStop(
```

```
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Methods

```
public System.String ToString();
```

Public Events

```
public void add_Enter(
   System.EventHandler value
);
public void remove_Enter(
   System.EventHandler value
);
public void add_KeyDown(
   System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyDown(
   System.Windows.Forms.KeyEventHandler value
);
public void add_KeyPress(
   System.Windows.Forms.KeyPressEventHandler value
);
public void remove_KeyPress(
   System.Windows.Forms.KeyPressEventHandler value
);
public void add_KeyUp(
   System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyUp(
   System.Windows.Forms.KeyEventHandler value
);
public void add_Leave(
   System.EventHandler value
);
public void remove_Leave(
   System.EventHandler value
);
public void add_SizeModeChanged(
   System.EventHandler value
);
public void remove_SizeModeChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Windows.Forms.ImeMode get_DefaultImeMode();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
```

```
);
protected void OnEnabledChanged(
   System.EventArgs e
);
protected void OnPaint(
   System.Windows.Forms.PaintEventArgs pe
);
protected void OnParentChanged(
   System.EventArgs e
);
protected void OnResize(
   System.EventArgs e
);
protected void OnSizeModeChanged(
   System.EventArgs e
);
protected void OnVisibleChanged(
   System.EventArgs e
);
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
```

PictureBoxSizeMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum PictureBoxSizeMode
```

Members

AutoSize

CenterImage

Normal

StretchImage

PrintControllerWithStatusDialog Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PrintControllerWithStatusDialog
   extends System.Drawing.Printing.PrintController
```

Public Constructors

```
public PrintControllerWithStatusDialog(
   System.Drawing.Printing.PrintController underlyingController
);
public PrintControllerWithStatusDialog(
   System.Drawing.Printing.PrintController underlyingController,
   System.String dialogTitle
);
```

Public Methods

```
public void OnEndPage(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintPageEventArgs e
);
public void OnEndPrint(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintEventArgs e
);
public System.Drawing.Graphics OnStartPage(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintPageEventArgs e
);
public void OnStartPrint(
   System.Drawing.Printing.PrintDocument document,
   System.Drawing.Printing.PrintEventArgs e
);
```

PrintDialog Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Document") */
public class PrintDialog
   extends System.Windows.Forms.CommonDialog
```

Public Constructors

```
public PrintDialog();
```

Public Properties

```
public boolean get_AllowPrintToFile();
public void set_AllowPrintToFile(
   boolean value
);
public boolean get_AllowSelection();
public void set_AllowSelection(
   boolean value
);
public boolean get_AllowSomePages();
public void set_AllowSomePages(
   boolean value
);
public System.Drawing.Printing.PrintDocument get_Document();
public void set_Document(
   System.Drawing.Printing.PrintDocument value
);
public System.Drawing.Printing.PrinterSettings get_PrinterSettings();
public void set_PrinterSettings(
   System.Drawing.Printing.PrinterSettings value
);
public boolean get_PrintToFile();
public void set_PrintToFile(
   boolean value
);
public boolean get_ShowHelp();
public void set_ShowHelp(
   boolean value
);
public boolean get_ShowNetwork();
public void set_ShowNetwork(
   boolean value
);
```

Public Methods

```
public void Reset();
```

Protected Methods

```
protected boolean RunDialog(
   System.IntPtr hwndOwner
```

```
);
```

PrintPreviewControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Document") */
public class PrintPreviewControl
   extends System.Windows.Forms.Control
```

Public Constructors

```
public PrintPreviewControl();
```

Public Properties

```
public boolean get_AutoZoom();
public void set_AutoZoom(
   boolean value
);
public int get_Columns();
public void set_Columns(
   int value
);
public System.Drawing.Printing.PrintDocument get_Document();
public void set_Document(
   System.Drawing.Printing.PrintDocument value
);
public int get_Rows();
public void set_Rows(
   int value
);
public int get_StartPage();
public void set_StartPage(
   int value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public boolean get_UseAntiAlias();
public void set_UseAntiAlias(
   boolean value
);
public double get_Zoom();
public void set_Zoom(
   double value
);
```

Public Methods

```
public void InvalidatePreview();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public void ResetBackColor();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Never) */
public void ResetForeColor();
```

Public Events

```
public void add_StartPageChanged(
   System.EventHandler value
);
public void remove_StartPageChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
```

Protected Methods

```
protected void OnPaint(
   System.Windows.Forms.PaintEventArgs pevent
);
protected void OnResize(
   System.EventArgs eventargs
);
protected void OnStartPageChanged(
   System.EventArgs e
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

PrintPreviewDialog Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.ComponentModel.Design.ComponentDesigner
, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemType = System.Drawing.De
sign.ToolboxItem, ToolboxItemTypeName = "System.Drawing.Design.ToolboxItem, System.Drawing, V
ersion=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Document") */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = true) */
public class PrintPreviewDialog
   extends System.Windows.Forms.Form
```

Public Constructors

```
public PrintPreviewDialog();
```

Public Properties

```
public System.Windows.Forms.IButtonControl get_AcceptButton();
public void set_AcceptButton(
   System.Windows.Forms.IButtonControl value
);
public System.String get_AccessibleDescription();
public void set_AccessibleDescription(
   System.String value
);
public System.String get_AccessibleName();
public void set_AccessibleName(
   System.String value
);
public System.Windows.Forms.AccessibleRole get_AccessibleRole();
public void set_AccessibleRole(
   System.Windows.Forms.AccessibleRole value
);
public boolean get_AllowDrop();
public void set_AllowDrop(
   boolean value
);
public System.Windows.Forms.AnchorStyles get_Anchor();
public void set_Anchor(
   System.Windows.Forms.AnchorStyles value
);
public boolean get_AutoScale();
public void set_AutoScale(
   boolean value
);
public System.Drawing.Size get_AutoScaleBaseSize();
public void set_AutoScaleBaseSize(
   System.Drawing.Size value
);
public boolean get_AutoScroll();
public void set_AutoScroll(
   boolean value
);
public System.Drawing.Size get_AutoScrollMargin();
public void set_AutoScrollMargin(
```

```
   System.Drawing.Size value
);
public System.Drawing.Size get_AutoScrollMinSize();
public void set_AutoScrollMinSize(
   System.Drawing.Size value
);
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.IButtonControl get_CancelButton();
public void set_CancelButton(
   System.Windows.Forms.IButtonControl value
);
public boolean get_CausesValidation();
public void set_CausesValidation(
   boolean value
);
public System.Windows.Forms.ContextMenu get_ContextMenu();
public void set_ContextMenu(
   System.Windows.Forms.ContextMenu value
);
public boolean get_ControlBox();
public void set_ControlBox(
   boolean value
);
public System.Windows.Forms.Cursor get_Cursor();
public void set_Cursor(
   System.Windows.Forms.Cursor value
);
public System.Windows.Forms.ControlBindingsCollection get_DataBindings();
public System.Windows.Forms.DockStyle get_Dock();
public void set_Dock(
   System.Windows.Forms.DockStyle value
);
public System.Windows.Forms.ScrollableControl+DockPaddingEdges get_DockPadding();
public System.Drawing.Printing.PrintDocument get_Document();
public void set_Document(
   System.Drawing.Printing.PrintDocument value
);
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.Windows.Forms.FormBorderStyle get_FormBorderStyle();
public void set_FormBorderStyle(
   System.Windows.Forms.FormBorderStyle value
);
public boolean get_HelpButton();
public void set_HelpButton(
   boolean value
);
public System.Drawing.Icon get_Icon();
public void set_Icon(
   System.Drawing.Icon value
);
```

```
public System.Windows.Forms.ImeMode get_ImeMode();
public void set_ImeMode(
   System.Windows.Forms.ImeMode value
);
public boolean get_IsMdiContainer();
public void set_IsMdiContainer(
   boolean value
);
public boolean get_KeyPreview();
public void set_KeyPreview(
   boolean value
);
public System.Drawing.Point get_Location();
public void set_Location(
   System.Drawing.Point value
);
public boolean get_MaximizeBox();
public void set_MaximizeBox(
   boolean value
);
public System.Drawing.Size get_MaximumSize();
public void set_MaximumSize(
   System.Drawing.Size value
);
public System.Windows.Forms.MainMenu get_Menu();
public void set_Menu(
   System.Windows.Forms.MainMenu value
);
public boolean get_MinimizeBox();
public void set_MinimizeBox(
   boolean value
);
public System.Drawing.Size get_MinimumSize();
public void set_MinimumSize(
   System.Drawing.Size value
);
public double get_Opacity();
public void set_Opacity(
   double value
);
public System.Windows.Forms.PrintPreviewControl get_PrintPreviewControl();
public System.Windows.Forms.RightToLeft get_RightToLeft();
public void set_RightToLeft(
   System.Windows.Forms.RightToLeft value
);
public boolean get_ShowInTaskbar();
public void set_ShowInTaskbar(
   boolean value
);
public System.Drawing.Size get_Size();
public void set_Size(
   System.Drawing.Size value
);
public System.Windows.Forms.SizeGripStyle get_SizeGripStyle();
public void set_SizeGripStyle(
   System.Windows.Forms.SizeGripStyle value
);
public System.Windows.Forms.FormStartPosition get_StartPosition();
public void set_StartPosition(
   System.Windows.Forms.FormStartPosition value
);
public boolean get_TabStop();
public void set_TabStop(
   boolean value
);
public System.Object get_Tag();
public void set_Tag(
   System.Object value
```

```
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public boolean get_TopMost();
public void set_TopMost(
   boolean value
);
public System.Drawing.Color get_TransparencyKey();
public void set_TransparencyKey(
   System.Drawing.Color value
);
public boolean get_UseAntiAlias();
public void set_UseAntiAlias(
   boolean value
);
public boolean get_Visible();
public void set_Visible(
   boolean value
);
public System.Windows.Forms.FormWindowState get_WindowState();
public void set_WindowState(
   System.Windows.Forms.FormWindowState value
);
```

Protected Methods

```
protected void CreateHandle();
protected void OnClosing(
   System.ComponentModel.CancelEventArgs e
);
```

ProgressBar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Value") */
public class ProgressBar
   extends System.Windows.Forms.Control
```

Public Constructors

```
public ProgressBar();
```

Public Properties

```
public boolean get_AllowDrop();
public void set_AllowDrop(
   boolean value
);
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public boolean get_CausesValidation();
public void set_CausesValidation(
   boolean value
);
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.Windows.Forms.ImeMode get_ImeMode();
public void set_ImeMode(
   System.Windows.Forms.ImeMode value
);
public int get_Maximum();
public void set_Maximum(
   int value
);
public int get_Minimum();
public void set_Minimum(
   int value
);
public System.Windows.Forms.RightToLeft get_RightToLeft();
public void set_RightToLeft(
   System.Windows.Forms.RightToLeft value
);
public int get_Step();
public void set_Step(
   int value
);
public boolean get_TabStop();
```

```
public void set_TabStop(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public int get_Value();
public void set_Value(
   int value
);
```

Public Methods

```
public void Increment(
   int value
);
public void PerformStep();
public System.String ToString();
```

Public Events

```
public void add_DoubleClick(
   System.EventHandler value
);
public void remove_DoubleClick(
   System.EventHandler value
);
public void add_Enter(
   System.EventHandler value
);
public void remove_Enter(
   System.EventHandler value
);
public void add_KeyDown(
   System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyDown(
   System.Windows.Forms.KeyEventHandler value
);
public void add_KeyPress(
   System.Windows.Forms.KeyPressEventHandler value
);
public void remove_KeyPress(
   System.Windows.Forms.KeyPressEventHandler value
);
public void add_KeyUp(
   System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyUp(
   System.Windows.Forms.KeyEventHandler value
);
public void add_Leave(
   System.EventHandler value
);
public void remove_Leave(
   System.EventHandler value
);
public void add_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Windows.Forms.ImeMode get_DefaultImeMode();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void CreateHandle();
protected void OnHandleCreated(
   System.EventArgs e
);
```

PropertyGrid Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.PropertyGridDesign
er, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class PropertyGrid
   extends System.Windows.Forms.ContainerControl
   implements System.Windows.Forms.ComponentModel.Com2Interop.IComPropertyBrowser
```

Public Constructors

```
public PropertyGrid();
```

Public Properties

```
public boolean get_AutoScroll();
public void set_AutoScroll(
   boolean value
);
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.ComponentModel.AttributeCollection get_BrowsableAttributes();
public void set_BrowsableAttributes(
   System.ComponentModel.AttributeCollection value
);
public boolean get_CanShowCommands();
public System.Drawing.Color get_CommandsBackColor();
public void set_CommandsBackColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_CommandsForeColor();
public void set_CommandsForeColor(
   System.Drawing.Color value
);
public boolean get_CommandsVisible();
public boolean get_CommandsVisibleIfAvailable();
public void set_CommandsVisibleIfAvailable(
   boolean value
);
public System.Drawing.Point get_ContextMenuDefaultLocation();
public System.Windows.Forms.Control+ControlCollection get_Controls();
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_HelpBackColor();
public void set_HelpBackColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_HelpForeColor();
public void set_HelpForeColor(
```

```
   System.Drawing.Color value
);
public boolean get_HelpVisible();
public void set_HelpVisible(
   boolean value
);
public boolean get_LargeButtons();
public void set_LargeButtons(
   boolean value
);
public System.Drawing.Color get_LineColor();
public void set_LineColor(
   System.Drawing.Color value
);
public System.Windows.Forms.PropertySort get_PropertySort();
public void set_PropertySort(
   System.Windows.Forms.PropertySort value
);
public System.Windows.Forms.PropertyGrid+PropertyTabCollection get_PropertyTabs();
public System.Windows.Forms.GridItem get_SelectedGridItem();
public void set_SelectedGridItem(
   System.Windows.Forms.GridItem value
);
public System.Object get_SelectedObject();
public void set_SelectedObject(
   System.Object value
);
public System.Object[] get_SelectedObjects();
public void set_SelectedObjects(
   System.Object[] value
);
public System.Windows.Forms.Design.PropertyTab get_SelectedTab();
public System.ComponentModel.ISite get_Site();
public void set_Site(
   System.ComponentModel.ISite value
);
public boolean get_ToolbarVisible();
public void set_ToolbarVisible(
   boolean value
);
public System.Drawing.Color get_ViewBackColor();
public void set_ViewBackColor(
   System.Drawing.Color value
);
public System.Drawing.Color get_ViewForeColor();
public void set_ViewForeColor(
   System.Drawing.Color value
);
```

Public Methods

```
public void CollapseAllGridItems();
public void ExpandAllGridItems();
public void Refresh();
public void RefreshTabs(
   System.ComponentModel.PropertyTabScope tabScope
);
public void ResetSelectedProperty();
```

Public Events

```
public void add_PropertySortChanged(
   System.EventHandler value
);
```

```
public void remove_PropertySortChanged(
   System.EventHandler value
);
public void add_PropertyTabChanged(
   System.Windows.Forms.PropertyTabChangedEventHandler value
);
public void remove_PropertyTabChanged(
   System.Windows.Forms.PropertyTabChangedEventHandler value
);
public void add_PropertyValueChanged(
   System.Windows.Forms.PropertyValueChangedEventHandler value
);
public void remove_PropertyValueChanged(
   System.Windows.Forms.PropertyValueChangedEventHandler value
);
public void add_SelectedGridItemChanged(
   System.Windows.Forms.SelectedGridItemChangedEventHandler value
);
public void remove_SelectedGridItemChanged(
   System.Windows.Forms.SelectedGridItemChangedEventHandler value
);
public void add_SelectedObjectsChanged(
   System.EventHandler value
);
public void remove_SelectedObjectsChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Drawing.Size get_DefaultSize();
protected System.Type get_DefaultTabType();
protected boolean get_DrawFlatToolbar();
protected void set_DrawFlatToolbar(
   boolean value
);
protected boolean get_ShowFocusCues();
```

Protected Methods

```
protected System.Windows.Forms.Design.PropertyTab CreatePropertyTab(
   System.Type tabType
);
protected void Dispose(
   boolean disposing
);
protected void OnComComponentNameChanged(
   System.ComponentModel.Design.ComponentRenameEventArgs e
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnGotFocus(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnHandleDestroyed(
   System.EventArgs e
);
protected void OnMouseDown(
   System.Windows.Forms.MouseEventArgs me
);
```

```
protected void OnMouseMove(
   System.Windows.Forms.MouseEventArgs me
);
protected void OnMouseUp(
   System.Windows.Forms.MouseEventArgs me
);
protected void OnNotifyPropertyValueUIItemsChanged(
   System.Object sender,
   System.EventArgs e
);
protected void OnPaint(
   System.Windows.Forms.PaintEventArgs pevent
);
protected void OnPropertyTabChanged(
   System.Windows.Forms.PropertyTabChangedEventArgs e
);
protected void OnPropertyValueChanged(
   System.Windows.Forms.PropertyValueChangedEventArgs e
);
protected void OnResize(
   System.EventArgs e
);
protected void OnSelectedGridItemChanged(
   System.Windows.Forms.SelectedGridItemChangedEventArgs e
);
protected void OnSelectedObjectsChanged(
   System.EventArgs e
);
protected void OnSystemColorsChanged(
   System.EventArgs e
);
protected void OnVisibleChanged(
   System.EventArgs e
);
protected boolean ProcessDialogKey(
   System.Windows.Forms.Keys keyData
);
protected void ScaleCore(
   float dx,
   float dy
);
protected void ShowEventsButton(
   boolean value
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

PropertyGrid.PropertyTabCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class PropertyTabCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Properties

```
public final int get_Count();
public System.Windows.Forms.Design.PropertyTab get_Item(
   int index
);
```

Public Methods

```
public void AddTabType(
   System.Type propertyTabType
);
public void AddTabType(
   System.Type propertyTabType,
   System.ComponentModel.PropertyTabScope tabScope
);
public void Clear(
   System.ComponentModel.PropertyTabScope tabScope
);
public final System.Collections.IEnumerator GetEnumerator();
public void RemoveTabType(
   System.Type propertyTabType
);
```

PropertyManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class PropertyManager
   extends System.Windows.Forms.BindingManagerBase
```

Public Constructors

```
public PropertyManager();
```

Public Properties

```
public int get_Count();
public System.Object get_Current();
public int get_Position();
public void set_Position(
   int value
);
```

Public Methods

```
public void AddNew();
public void CancelCurrentEdit();
public void EndCurrentEdit();
public System.ComponentModel.PropertyDescriptorCollection GetItemProperties();
public void RemoveAt(
   int index
);
public void ResumeBinding();
public void SuspendBinding();
```

Protected Methods

```
protected System.String GetListName(
   System.Collections.ArrayList listAccessors
);
protected void OnCurrentChanged(
   System.EventArgs ea
);
protected void UpdateIsBinding();
```

PropertySort Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum PropertySort
```

Members

Alphabetical

Categorized

CategorizedAlphabetical

NoSort

PropertyTabChangedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class PropertyTabChangedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public PropertyTabChangedEventArgs(
   System.Windows.Forms.Design.PropertyTab oldTab,
   System.Windows.Forms.Design.PropertyTab newTab
);
```

Public Properties

```
public System.Windows.Forms.Design.PropertyTab get_NewTab();
public System.Windows.Forms.Design.PropertyTab get_OldTab();
```

PropertyTabChangedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void PropertyTabChangedEventHandler(
   System.Object s,
   System.Windows.Forms.PropertyTabChangedEventArgs e
);
```

PropertyValueChangedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class PropertyValueChangedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public PropertyValueChangedEventArgs(
   System.Windows.Forms.GridItem changedItem,
   System.Object oldValue
);
```

Public Properties

```
public System.Windows.Forms.GridItem get_ChangedItem();
public System.Object get_OldValue();
```

PropertyValueChangedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void PropertyValueChangedEventHandler(
   System.Object s,
   System.Windows.Forms.PropertyValueChangedEventArgs e
);
```

QueryAccessibilityHelpEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class QueryAccessibilityHelpEventArgs
   extends System.EventArgs
```

Public Constructors

```
public QueryAccessibilityHelpEventArgs();
public QueryAccessibilityHelpEventArgs(
   System.String helpNamespace,
   System.String helpString,
   System.String helpKeyword
);
```

Public Properties

```
public System.String get_HelpKeyword();
public void set_HelpKeyword(
   System.String value
);
public System.String get_HelpNamespace();
public void set_HelpNamespace(
   System.String value
);
public System.String get_HelpString();
public void set_HelpString(
   System.String value
);
```

QueryAccessibilityHelpEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void QueryAccessibilityHelpEventHandler(
   System.Object sender,
   System.Windows.Forms.QueryAccessibilityHelpEventArgs e
);
```

QueryContinueDragEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class QueryContinueDragEventArgs
   extends System.EventArgs
```

Public Constructors

```
public QueryContinueDragEventArgs(
   int keyState,
   boolean escapePressed,
   System.Windows.Forms.DragAction action
);
```

Public Properties

```
public System.Windows.Forms.DragAction get_Action();
public void set_Action(
   System.Windows.Forms.DragAction value
);
public boolean get_EscapePressed();
public int get_KeyState();
```

QueryContinueDragEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void QueryContinueDragEventHandler(
   System.Object sender,
   System.Windows.Forms.QueryContinueDragEventArgs e
);
```

RadioButton Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "CheckedChanged") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Checked") */
public class RadioButton
   extends System.Windows.Forms.ButtonBase
```

Public Constructors

```
public RadioButton();
```

Public Properties

```
public System.Windows.Forms.Appearance get_Appearance();
public void set_Appearance(
   System.Windows.Forms.Appearance value
);
public boolean get_AutoCheck();
public void set_AutoCheck(
   boolean value
);
public System.Drawing.ContentAlignment get_CheckAlign();
public void set_CheckAlign(
   System.Drawing.ContentAlignment value
);
public boolean get_Checked();
public void set_Checked(
   boolean value
);
public boolean get_TabStop();
public void set_TabStop(
   boolean value
);
public System.Drawing.ContentAlignment get_TextAlign();
public void set_TextAlign(
   System.Drawing.ContentAlignment value
);
```

Public Methods

```
public void PerformClick();
public System.String ToString();
```

Public Events

```
public void add_AppearanceChanged(
   System.EventHandler value
);
public void remove_AppearanceChanged(
   System.EventHandler value
);
public void add_CheckedChanged(
   System.EventHandler value
```

```
);
public void remove_CheckedChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected System.Windows.Forms.AccessibleObject CreateAccessibilityInstance();
protected void OnCheckedChanged(
   System.EventArgs e
);
protected void OnClick(
   System.EventArgs e
);
protected void OnEnter(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnMouseUp(
   System.Windows.Forms.MouseEventArgs mevent
);
protected boolean ProcessMnemonic(
   char charCode
);
```

RadioButton.RadioButtonAccessibleObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class RadioButtonAccessibleObject
   extends System.Windows.Forms.Control+ControlAccessibleObject
```

Public Constructors

```
public RadioButtonAccessibleObject(
   System.Windows.Forms.RadioButton owner
);
```

Public Properties

```
public System.String get_DefaultAction();
public System.Windows.Forms.AccessibleRole get_Role();
public System.Windows.Forms.AccessibleStates get_State();
```

Public Methods

```
public void DoDefaultAction();
```

RichTextBox Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class RichTextBox
   extends System.Windows.Forms.TextBoxBase
```

Public Constructors

```
public RichTextBox();
```

Public Properties

```
public boolean get_AllowDrop();
public void set_AllowDrop(
   boolean value
);
public boolean get_AutoSize();
public void set_AutoSize(
   boolean value
);
public boolean get_AutoWordSelection();
public void set_AutoWordSelection(
   boolean value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public int get_BulletIndent();
public void set_BulletIndent(
   int value
);
public boolean get_CanRedo();
public boolean get_DetectUrls();
public void set_DetectUrls(
   boolean value
);
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public int get_MaxLength();
public void set_MaxLength(
   int value
);
public boolean get_Multiline();
public void set_Multiline(
   boolean value
);
public System.String get_RedoActionName();
public int get_RightMargin();
public void set_RightMargin(
   int value
);
```

```
public System.String get_Rtf();
public void set_Rtf(
   System.String value
);
public System.Windows.Forms.RichTextBoxScrollBars get_ScrollBars();
public void set_ScrollBars(
   System.Windows.Forms.RichTextBoxScrollBars value
);
public System.String get_SelectedRtf();
public void set_SelectedRtf(
   System.String value
);
public System.String get_SelectedText();
public void set_SelectedText(
   System.String value
);
public System.Windows.Forms.HorizontalAlignment get_SelectionAlignment();
public void set_SelectionAlignment(
   System.Windows.Forms.HorizontalAlignment value
);
public boolean get_SelectionBullet();
public void set_SelectionBullet(
   boolean value
);
public int get_SelectionCharOffset();
public void set_SelectionCharOffset(
   int value
);
public System.Drawing.Color get_SelectionColor();
public void set_SelectionColor(
   System.Drawing.Color value
);
public System.Drawing.Font get_SelectionFont();
public void set_SelectionFont(
   System.Drawing.Font value
);
public int get_SelectionHangingIndent();
public void set_SelectionHangingIndent(
   int value
);
public int get_SelectionIndent();
public void set_SelectionIndent(
   int value
);
public int get_SelectionLength();
public void set_SelectionLength(
   int value
);
public boolean get_SelectionProtected();
public void set_SelectionProtected(
   boolean value
);
public int get_SelectionRightIndent();
public void set_SelectionRightIndent(
   int value
);
public int[] get_SelectionTabs();
public void set_SelectionTabs(
   int[] value
);
public System.Windows.Forms.RichTextBoxSelectionTypes get_SelectionType();
public boolean get_ShowSelectionMargin();
public void set_ShowSelectionMargin(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
```

```
);
public int get_TextLength();
public System.String get_UndoActionName();
public float get_ZoomFactor();
public void set_ZoomFactor(
   float value
);
```

Public Methods

```
public boolean CanPaste(
   System.Windows.Forms.DataFormats+Format clipFormat
);
public int Find(
   char[] characterSet
);
public int Find(
   System.String str
);
public int Find(
   char[] characterSet,
   int start
);
public int Find(
   System.String str,
   System.Windows.Forms.RichTextBoxFinds options
);
public int Find(
   char[] characterSet,
   int start,
   int end
);
public int Find(
   System.String str,
   int start,
   System.Windows.Forms.RichTextBoxFinds options
);
public int Find(
   System.String str,
   int start,
   int end,
   System.Windows.Forms.RichTextBoxFinds options
);
public char GetCharFromPosition(
   System.Drawing.Point pt
);
public int GetCharIndexFromPosition(
   System.Drawing.Point pt
);
public int GetLineFromCharIndex(
   int index
);
public System.Drawing.Point GetPositionFromCharIndex(
   int index
);
public void LoadFile(
   System.String path
);
public void LoadFile(
   System.IO.Stream data,
   System.Windows.Forms.RichTextBoxStreamType fileType
);
public void LoadFile(
   System.String path,
   System.Windows.Forms.RichTextBoxStreamType fileType
);
```

```
public void Paste(
   System.Windows.Forms.DataFormats+Format clipFormat
);
public void Redo();
public void SaveFile(
   System.String path
);
public void SaveFile(
   System.IO.Stream data,
   System.Windows.Forms.RichTextBoxStreamType fileType
);
public void SaveFile(
   System.String path,
   System.Windows.Forms.RichTextBoxStreamType fileType
);
```

Public Events

```
public void add_ContentsResized(
   System.Windows.Forms.ContentsResizedEventHandler value
);
public void remove_ContentsResized(
   System.Windows.Forms.ContentsResizedEventHandler value
);
public void add_DoubleClick(
   System.EventHandler value
);
public void remove_DoubleClick(
   System.EventHandler value
);
public void add_DragDrop(
   System.Windows.Forms.DragEventHandler value
);
public void remove_DragDrop(
   System.Windows.Forms.DragEventHandler value
);
public void add_DragEnter(
   System.Windows.Forms.DragEventHandler value
);
public void remove_DragEnter(
   System.Windows.Forms.DragEventHandler value
);
public void add_DragLeave(
   System.EventHandler value
);
public void remove_DragLeave(
   System.EventHandler value
);
public void add_DragOver(
   System.Windows.Forms.DragEventHandler value
);
public void remove_DragOver(
   System.Windows.Forms.DragEventHandler value
);
public void add_GiveFeedback(
   System.Windows.Forms.GiveFeedbackEventHandler value
);
public void remove_GiveFeedback(
   System.Windows.Forms.GiveFeedbackEventHandler value
);
public void add_HScroll(
   System.EventHandler value
);
public void remove_HScroll(
   System.EventHandler value
);
```

```
public void add_ImeChange(
   System.EventHandler value
);
public void remove_ImeChange(
   System.EventHandler value
);
public void add_LinkClicked(
   System.Windows.Forms.LinkClickedEventHandler value
);
public void remove_LinkClicked(
   System.Windows.Forms.LinkClickedEventHandler value
);
public void add_Protected(
   System.EventHandler value
);
public void remove_Protected(
   System.EventHandler value
);
public void add_QueryContinueDrag(
   System.Windows.Forms.QueryContinueDragEventHandler value
);
public void remove_QueryContinueDrag(
   System.Windows.Forms.QueryContinueDragEventHandler value
);
public void add_SelectionChanged(
   System.EventHandler value
);
public void remove_SelectionChanged(
   System.EventHandler value
);
public void add_VScroll(
   System.EventHandler value
);
public void remove_VScroll(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected System.Object CreateRichEditOleCallback();
protected void OnBackColorChanged(
   System.EventArgs e
);
protected void OnContentsResized(
   System.Windows.Forms.ContentsResizedEventArgs e
);
protected void OnContextMenuChanged(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnHandleDestroyed(
   System.EventArgs e
);
protected void OnHScroll(
   System.EventArgs e
);
protected void OnImeChange(
```

```
   System.EventArgs e
);
protected void OnLinkClicked(
   System.Windows.Forms.LinkClickedEventArgs e
);
protected void OnProtected(
   System.EventArgs e
);
protected void OnRightToLeftChanged(
   System.EventArgs e
);
protected void OnSelectionChanged(
   System.EventArgs e
);
protected void OnSystemColorsChanged(
   System.EventArgs e
);
protected void OnTextChanged(
   System.EventArgs e
);
protected void OnVScroll(
   System.EventArgs e
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

RichTextBoxFinds Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum RichTextBoxFinds
```

Members

MatchCase

NoHighlight

None

Reverse

WholeWord

RichTextBoxScrollBars Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum RichTextBoxScrollBars
```

Members

Both

ForcedBoth

ForcedHorizontal

ForcedVertical

Horizontal

None

Vertical

RichTextBoxSelectionAttribute Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum RichTextBoxSelectionAttribute
```

Members

All

Mixed

None

RichTextBoxSelectionTypes Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum RichTextBoxSelectionTypes
```

Members

Empty

MultiChar

MultiObject

Object

Text

RichTextBoxStreamType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum RichTextBoxStreamType
```

Members

PlainText

RichNoOleObjs

RichText

TextTextOleObjs

UnicodePlainText

RichTextBoxWordPunctuations Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum RichTextBoxWordPunctuations
```

Members

All

Custom

Level1

Level2

RightToLeft Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum RightToLeft
```

Members

Inherit

No

Yes

SaveFileDialog Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.SaveFileDialogDesi
gner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") *
/
public class SaveFileDialog
   extends System.Windows.Forms.FileDialog
```

Public Constructors

```
public SaveFileDialog();
```

Public Properties

```
public boolean get_CreatePrompt();
public void set_CreatePrompt(
   boolean value
);
public boolean get_OverwritePrompt();
public void set_OverwritePrompt(
   boolean value
);
```

Public Methods

```
public System.IO.Stream OpenFile();
public void Reset();
```

Screen Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Screen
   extends System.Object
```

Public Properties

```
public static System.Windows.Forms.Screen[] get_AllScreens();
public System.Drawing.Rectangle get_Bounds();
public System.String get_DeviceName();
public boolean get_Primary();
public static System.Windows.Forms.Screen get_PrimaryScreen();
public System.Drawing.Rectangle get_WorkingArea();
```

Public Methods

```
public boolean Equals(
   System.Object obj
);
public static System.Windows.Forms.Screen FromControl(
   System.Windows.Forms.Control control
);
public static System.Windows.Forms.Screen FromHandle(
   System.IntPtr hwnd
);
public static System.Windows.Forms.Screen FromPoint(
   System.Drawing.Point point
);
public static System.Windows.Forms.Screen FromRectangle(
   System.Drawing.Rectangle rect
);
public static System.Drawing.Rectangle GetBounds(
   System.Drawing.Point pt
);
public static System.Drawing.Rectangle GetBounds(
   System.Drawing.Rectangle rect
);
public static System.Drawing.Rectangle GetBounds(
   System.Windows.Forms.Control ctl
);
public int GetHashCode();
public static System.Drawing.Rectangle GetWorkingArea(
   System.Drawing.Point pt
);
public static System.Drawing.Rectangle GetWorkingArea(
   System.Drawing.Rectangle rect
);
public static System.Drawing.Rectangle GetWorkingArea(
   System.Windows.Forms.Control ctl
);
public System.String ToString();
```

ScrollableControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.ScrollableControlD
esigner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
) */
public class ScrollableControl
   extends System.Windows.Forms.Control
```

Public Constructors

```
public ScrollableControl();
```

Public Properties

```
public boolean get_AutoScroll();
public void set_AutoScroll(
   boolean value
);
public System.Drawing.Size get_AutoScrollMargin();
public void set_AutoScrollMargin(
   System.Drawing.Size value
);
public System.Drawing.Size get_AutoScrollMinSize();
public void set_AutoScrollMinSize(
   System.Drawing.Size value
);
public System.Drawing.Point get_AutoScrollPosition();
public void set_AutoScrollPosition(
   System.Drawing.Point value
);
public System.Drawing.Rectangle get_DisplayRectangle();
public System.Windows.Forms.ScrollableControl+DockPaddingEdges get_DockPadding();
```

Public Methods

```
public void ScrollControlIntoView(
   System.Windows.Forms.Control activeControl
);
public void SetAutoScrollMargin(
   int x,
   int y
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected boolean get_HScroll();
protected void set_HScroll(
   boolean value
);
protected boolean get_VScroll();
protected void set_VScroll(
```

```
   boolean value
);
```

Protected Methods

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void AdjustFormScrollbars(
   boolean displayScrollbars
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected boolean GetScrollState(
   int bit
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnLayout(
   System.Windows.Forms.LayoutEventArgs levent
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMouseWheel(
   System.Windows.Forms.MouseEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnVisibleChanged(
   System.EventArgs e
);
protected void ScaleCore(
   float dx,
   float dy
);
protected void SetDisplayRectLocation(
   int x,
   int y
);
protected void SetScrollState(
   int bit,
   boolean value
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

Protected Fields

```
protected static int ScrollStateAutoScrolling;
protected static int ScrollStateFullDrag;
protected static int ScrollStateHScrollVisible;
protected static int ScrollStateUserHasScrolled;
protected static int ScrollStateVScrollVisible;
```

ScrollableControl.DockPaddingEdges Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Windo
ws.Forms.ScrollableControl+DockPaddingEdgesConverter, System.Windows.Forms, Version=1.0.3300.
0, Culture=neutral, PublicKeyToken=b77a5c561934e089") */
public class DockPaddingEdges
   extends System.Object
   implements System.ICloneable
```

Public Properties

```
public int get_All();
public void set_All(
   int value
);
public int get_Bottom();
public void set_Bottom(
   int value
);
public int get_Left();
public void set_Left(
   int value
);
public int get_Right();
public void set_Right(
   int value
);
public int get_Top();
public void set_Top(
   int value
);
```

Public Methods

```
public boolean Equals(
   System.Object other
);
public int GetHashCode();
public System.String ToString();
```

ScrollableControl.DockPaddingEdgesConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DockPaddingEdgesConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public DockPaddingEdgesConverter();
```

Public Methods

```
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value,
   System.Attribute[] attributes
);
public boolean GetPropertiesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

ScrollBar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Value") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Scroll") */
public abstract class ScrollBar
   extends System.Windows.Forms.Control
```

Public Constructors

```
public ScrollBar();
```

Public Properties

```
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.Windows.Forms.ImeMode get_ImeMode();
public void set_ImeMode(
   System.Windows.Forms.ImeMode value
);
public int get_LargeChange();
public void set_LargeChange(
   int value
);
public int get_Maximum();
public void set_Maximum(
   int value
);
public int get_Minimum();
public void set_Minimum(
   int value
);
public int get_SmallChange();
public void set_SmallChange(
   int value
);
public boolean get_TabStop();
public void set_TabStop(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

```
public int get_Value();
public void set_Value(
   int value
);
```

Public Methods

```
public System.String ToString();
```

Public Events

```
public void add_Click(
   System.EventHandler value
);
public void remove_Click(
   System.EventHandler value
);
public void add_DoubleClick(
   System.EventHandler value
);
public void remove_DoubleClick(
   System.EventHandler value
);
public void add_MouseDown(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseDown(
   System.Windows.Forms.MouseEventHandler value
);
public void add_MouseMove(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseMove(
   System.Windows.Forms.MouseEventHandler value
);
public void add_MouseUp(
   System.Windows.Forms.MouseEventHandler value
);
public void remove_MouseUp(
   System.Windows.Forms.MouseEventHandler value
);
public void add_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void add_Scroll(
   System.Windows.Forms.ScrollEventHandler value
);
public void remove_Scroll(
   System.Windows.Forms.ScrollEventHandler value
);
public void add_ValueChanged(
   System.EventHandler value
);
public void remove_ValueChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Windows.Forms.ImeMode get_DefaultImeMode();
```

Protected Methods

```
protected void OnEnabledChanged(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnScroll(
   System.Windows.Forms.ScrollEventArgs se
);
protected void OnValueChanged(
   System.EventArgs e
);
protected void UpdateScrollInfo();
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

ScrollBars Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ScrollBars
```

Members

Both

Horizontal

None

Vertical

ScrollButton Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ScrollButton
```

Members

Down

Left

Max

Min

Right

Up

ScrollEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class ScrollEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ScrollEventArgs(
   System.Windows.Forms.ScrollEventType type,
   int newValue
);
```

Public Properties

```
public int get_NewValue();
public void set_NewValue(
   int value
);
public System.Windows.Forms.ScrollEventType get_Type();
```

ScrollEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ScrollEventHandler(
   System.Object sender,
   System.Windows.Forms.ScrollEventArgs e
);
```

ScrollEventType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum ScrollEventType
```

Members

EndScroll

First

LargeDecrement

LargeIncrement

Last

SmallDecrement

SmallIncrement

ThumbPosition

ThumbTrack

SecurityIDType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SecurityIDType
```

Members

Alias

Computer

DeletedAccount

Domain

Group

Invalid

Unknown

User

WellKnownGroup

SelectedGridItemChangedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SelectedGridItemChangedEventArgs
   extends System.EventArgs
```

Public Constructors

```
public SelectedGridItemChangedEventArgs(
   System.Windows.Forms.GridItem oldSel,
   System.Windows.Forms.GridItem newSel
);
```

Public Properties

```
public System.Windows.Forms.GridItem get_NewSelection();
public System.Windows.Forms.GridItem get_OldSelection();
```

SelectedGridItemChangedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void SelectedGridItemChangedEventHandler(
   System.Object sender,
   System.Windows.Forms.SelectedGridItemChangedEventArgs e
);
```

SelectionMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum SelectionMode
```

Members

MultiExtended

MultiSimple

None

One

SelectionRange Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Windo
ws.Forms.SelectionRangeConverter, System.Windows.Forms, Version=1.0.3300.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089") */
public class SelectionRange
   extends System.Object
```

Public Constructors

```
public SelectionRange();
public SelectionRange(
   System.Windows.Forms.SelectionRange range
);
public SelectionRange(
   System.DateTime lower,
   System.DateTime upper
);
```

Public Properties

```
public System.DateTime get_End();
public void set_End(
   System.DateTime value
);
public System.DateTime get_Start();
public void set_Start(
   System.DateTime value
);
```

Public Methods

```
public System.String ToString();
```

SelectionRangeConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SelectionRangeConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public SelectionRangeConverter();
```

Public Methods

```
public boolean CanConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type sourceType
);
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertFrom(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
public System.Object CreateInstance(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Collections.IDictionary propertyValues
);
public boolean GetCreateInstanceSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object value,
   System.Attribute[] attributes
);
public boolean GetPropertiesSupported(
   System.ComponentModel.ITypeDescriptorContext context
);
```

SendKeys Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SendKeys
   extends System.Object
```

Public Methods

```
public static void Flush();
public static void Send(
   System.String keys
);
public static void SendWait(
   System.String keys
);
```

Shortcut Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public enum Shortcut
```

Members

Alt0

Alt1

Alt2

Alt3

Alt4

Alt5

Alt6

Alt7

Alt8

Alt9

AltBksp

AltF1

AltF10

AltF11

AltF12

AltF2

AltF3

AltF4

AltF5

AltF6

AltF7

AltF8

AltF9

Ctrl0

Ctrl1

Ctrl2

Ctrl3

Ctrl4

Ctrl5

Ctrl6

Ctrl7

Ctrl8

Ctrl9

CtrlA

CtrlB

CtrlC

CtrlD

CtrlDel

CtrlE

CtrlF

CtrlF1

CtrlF10

CtrlF11

CtrlF12

CtrlF2

CtrlF3

CtrlF4

CtrlF5

CtrlF6

CtrlF7

CtrlF8

CtrlF9

CtrlG

CtrlH

CtrlI

CtrlIns

CtrlJ

CtrlK

CtrlL

CtrlM

CtrlN

CtrlO

CtrlP

CtrlQ

CtrlR

CtrlS

CtrlShift0

CtrlShift1

CtrlShift2

CtrlShift3

CtrlShift4

CtrlShift5

CtrlShift6

CtrlShift7

CtrlShift8

CtrlShift9

CtrlShiftA

CtrlShiftB

CtrlShiftC

CtrlShiftD

CtrlShiftE

CtrlShiftF

CtrlShiftF1

CtrlShiftF10

CtrlShiftF11

CtrlShiftF12

CtrlShiftF2

CtrlShiftF3

CtrlShiftF4

CtrlShiftF5

CtrlShiftF6

CtrlShiftF7

CtrlShiftF8

CtrlShiftF9

CtrlShiftG

CtrlShiftH

CtrlShiftI

CtrlShiftJ

CtrlShiftK

CtrlShiftL

CtrlShiftM

CtrlShiftN

CtrlShiftO

CtrlShiftP

CtrlShiftQ

CtrlShiftR

CtrlShiftS

CtrlShiftT

CtrlShiftU

CtrlShiftV

CtrlShiftW

CtrlShiftX

CtrlShiftY

CtrlShiftZ

CtrlT

CtrlU

CtrlV

CtrlW

CtrlX

CtrlY

CtrlZ

Del

F1

F10

F11

F12

F2

F3

F4

F5

F6

F7

F8

F9

Ins

None

ShiftDel

ShiftF1

ShiftF10

ShiftF11

ShiftF12

ShiftF2

ShiftF3

ShiftF4

ShiftF5

ShiftF6

ShiftF7

ShiftF8

ShiftF9

ShiftIns

SizeGripStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SizeGripStyle
```

Members

Auto

Hide

Show

SortOrder Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum SortOrder
```

Members

Ascending

Descending

None

Splitter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Dock") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.SplitterDesigner,
System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "SplitterMoved") */
public class Splitter
   extends System.Windows.Forms.Control
   implements System.Windows.Forms.IMessageFilter
```

Public Constructors

```
public Splitter();
```

Public Properties

```
public boolean get_AllowDrop();
public void set_AllowDrop(
   boolean value
);
public System.Windows.Forms.AnchorStyles get_Anchor();
public void set_Anchor(
   System.Windows.Forms.AnchorStyles value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Windows.Forms.BorderStyle value
);
public System.Windows.Forms.DockStyle get_Dock();
public void set_Dock(
   System.Windows.Forms.DockStyle value
);
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.Windows.Forms.ImeMode get_ImeMode();
public void set_ImeMode(
   System.Windows.Forms.ImeMode value
);
public int get_MinExtra();
public void set_MinExtra(
   int value
);
public int get_MinSize();
public void set_MinSize(
   int value
);
```

```
public int get_SplitPosition();
public void set_SplitPosition(
   int value
);
public boolean get_TabStop();
public void set_TabStop(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Methods

```
public final boolean PreFilterMessage(
   System.Windows.Forms.Message& m
);
public System.String ToString();
```

Public Events

```
public void add_Enter(
   System.EventHandler value
);
public void remove_Enter(
   System.EventHandler value
);
public void add_KeyDown(
   System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyDown(
   System.Windows.Forms.KeyEventHandler value
);
public void add_KeyPress(
   System.Windows.Forms.KeyPressEventHandler value
);
public void remove_KeyPress(
   System.Windows.Forms.KeyPressEventHandler value
);
public void add_KeyUp(
   System.Windows.Forms.KeyEventHandler value
);
public void remove_KeyUp(
   System.Windows.Forms.KeyEventHandler value
);
public void add_Leave(
   System.EventHandler value
);
public void remove_Leave(
   System.EventHandler value
);
public void add_SplitterMoved(
   System.Windows.Forms.SplitterEventHandler value
);
public void remove_SplitterMoved(
   System.Windows.Forms.SplitterEventHandler value
);
public void add_SplitterMoving(
   System.Windows.Forms.SplitterEventHandler value
);
public void remove_SplitterMoving(
   System.Windows.Forms.SplitterEventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Windows.Forms.ImeMode get_DefaultImeMode();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void OnKeyDown(
   System.Windows.Forms.KeyEventArgs e
);
protected void OnMouseDown(
   System.Windows.Forms.MouseEventArgs e
);
protected void OnMouseMove(
   System.Windows.Forms.MouseEventArgs e
);
protected void OnMouseUp(
   System.Windows.Forms.MouseEventArgs e
);
protected void OnSplitterMoved(
   System.Windows.Forms.SplitterEventArgs sevent
);
protected void OnSplitterMoving(
   System.Windows.Forms.SplitterEventArgs sevent
);
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
```

SplitterEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class SplitterEventArgs
   extends System.EventArgs
```

Public Constructors

```
public SplitterEventArgs(
   int x,
   int y,
   int splitX,
   int splitY
);
```

Public Properties

```
public int get_SplitX();
public void set_SplitX(
   int value
);
public int get_SplitY();
public void set_SplitY(
   int value
);
public int get_X();
public int get_Y();
```

SplitterEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void SplitterEventHandler(
   System.Object sender,
   System.Windows.Forms.SplitterEventArgs e
);
```

StatusBar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "PanelClick") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.StatusBarDesigner,
System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class StatusBar
   extends System.Windows.Forms.Control
```

Public Constructors

```
public StatusBar();
```

Public Properties

```
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.DockStyle get_Dock();
public void set_Dock(
   System.Windows.Forms.DockStyle value
);
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.Windows.Forms.ImeMode get_ImeMode();
public void set_ImeMode(
   System.Windows.Forms.ImeMode value
);
public System.Windows.Forms.StatusBar+StatusBarPanelCollection get_Panels();
public boolean get_ShowPanels();
public void set_ShowPanels(
   boolean value
);
public boolean get_SizingGrip();
public void set_SizingGrip(
   boolean value
);
public boolean get_TabStop();
public void set_TabStop(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Methods

```
public System.String ToString();
```

Public Events

```
public void add_DrawItem(
   System.Windows.Forms.StatusBarDrawItemEventHandler value
);
public void remove_DrawItem(
   System.Windows.Forms.StatusBarDrawItemEventHandler value
);
public void add_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void add_PanelClick(
   System.Windows.Forms.StatusBarPanelClickEventHandler value
);
public void remove_PanelClick(
   System.Windows.Forms.StatusBarPanelClickEventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Windows.Forms.ImeMode get_DefaultImeMode();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void CreateHandle();
protected void Dispose(
   boolean disposing
);
protected void OnDrawItem(
   System.Windows.Forms.StatusBarDrawItemEventArgs sbdievent
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnHandleDestroyed(
   System.EventArgs e
);
protected void OnLayout(
   System.Windows.Forms.LayoutEventArgs levent
);
protected void OnMouseDown(
   System.Windows.Forms.MouseEventArgs e
);
protected void OnPanelClick(
   System.Windows.Forms.StatusBarPanelClickEventArgs e
);
protected void OnResize(
   System.EventArgs e
);
protected void WndProc(
```

```
    System.Windows.Forms.Message& m
);
```

StatusBar.StatusBarPanelCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class StatusBarPanelCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public StatusBarPanelCollection(
   System.Windows.Forms.StatusBar owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public System.Windows.Forms.StatusBarPanel get_Item(
   int index
);
public void set_Item(
   int index,
   System.Windows.Forms.StatusBarPanel value
);
```

Public Methods

```
public int Add(
   System.Windows.Forms.StatusBarPanel value
);
public System.Windows.Forms.StatusBarPanel Add(
   System.String text
);
public void AddRange(
   System.Windows.Forms.StatusBarPanel[] panels
);
public void Clear();
public boolean Contains(
   System.Windows.Forms.StatusBarPanel panel
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Windows.Forms.StatusBarPanel panel
);
public void Insert(
   int index,
   System.Windows.Forms.StatusBarPanel value
);
public void Remove(
   System.Windows.Forms.StatusBarPanel value
);
public void RemoveAt(
   int index
);
```

StatusBarDrawItemEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class StatusBarDrawItemEventArgs
   extends System.Windows.Forms.DrawItemEventArgs
```

Public Constructors

```
public StatusBarDrawItemEventArgs(
   System.Drawing.Graphics g,
   System.Drawing.Font font,
   System.Drawing.Rectangle r,
   int itemId,
   System.Windows.Forms.DrawItemState itemState,
   System.Windows.Forms.StatusBarPanel panel
);
```

Public Properties

```
public System.Windows.Forms.StatusBarPanel get_Panel();
```

StatusBarDrawItemEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void StatusBarDrawItemEventHandler(
   System.Object sender,
   System.Windows.Forms.StatusBarDrawItemEventArgs sbdevent
);
```

StatusBarPanel Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
public class StatusBarPanel
   extends System.ComponentModel.Component
   implements System.ComponentModel.ISupportInitialize
```

Public Constructors

```
public StatusBarPanel();
```

Public Properties

```
public System.Windows.Forms.HorizontalAlignment get_Alignment();
public void set_Alignment(
   System.Windows.Forms.HorizontalAlignment value
);
public System.Windows.Forms.StatusBarPanelAutoSize get_AutoSize();
public void set_AutoSize(
   System.Windows.Forms.StatusBarPanelAutoSize value
);
public System.Windows.Forms.StatusBarPanelBorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Windows.Forms.StatusBarPanelBorderStyle value
);
public System.Drawing.Icon get_Icon();
public void set_Icon(
   System.Drawing.Icon value
);
public int get_MinWidth();
public void set_MinWidth(
   int value
);
public System.Windows.Forms.StatusBar get_Parent();
public System.Windows.Forms.StatusBarPanelStyle get_Style();
public void set_Style(
   System.Windows.Forms.StatusBarPanelStyle value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.String get_ToolTipText();
public void set_ToolTipText(
   System.String value
);
public int get_Width();
public void set_Width(
   int value
);
```

Public Methods

```
public final void BeginInit();
```

```
public final void EndInit();
public System.String ToString();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

StatusBarPanelAutoSize Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum StatusBarPanelAutoSize
```

Members

Contents

None

Spring

StatusBarPanelBorderStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum StatusBarPanelBorderStyle
```

Members

None

Raised

Sunken

StatusBarPanelClickEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class StatusBarPanelClickEventArgs
   extends System.Windows.Forms.MouseEventArgs
```

Public Constructors

```
public StatusBarPanelClickEventArgs(
   System.Windows.Forms.StatusBarPanel statusBarPanel,
   System.Windows.Forms.MouseButtons button,
   int clicks,
   int x,
   int y
);
```

Public Properties

```
public System.Windows.Forms.StatusBarPanel get_StatusBarPanel();
```

StatusBarPanelClickEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void StatusBarPanelClickEventHandler(
   System.Object sender,
   System.Windows.Forms.StatusBarPanelClickEventArgs e
);
```

StatusBarPanelStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum StatusBarPanelStyle
```

Members

OwnerDraw

Text

StructFormat Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum StructFormat
```

Members

Ansi

Auto

Unicode

SystemInformation Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SystemInformation
   extends System.Object
```

Public Properties

```
public static System.Windows.Forms.ArrangeDirection get_ArrangeDirection();
public static System.Windows.Forms.ArrangeStartingPosition get_ArrangeStartingPosition();
public static System.Windows.Forms.BootMode get_BootMode();
public static System.Drawing.Size get_Border3DSize();
public static System.Drawing.Size get_BorderSize();
public static System.Drawing.Size get_CaptionButtonSize();
public static int get_CaptionHeight();
public static System.String get_ComputerName();
public static System.Drawing.Size get_CursorSize();
public static boolean get_DbcsEnabled();
public static boolean get_DebugOS();
public static System.Drawing.Size get_DoubleClickSize();
public static int get_DoubleClickTime();
public static boolean get_DragFullWindows();
public static System.Drawing.Size get_DragSize();
public static System.Drawing.Size get_FixedFrameBorderSize();
public static System.Drawing.Size get_FrameBorderSize();
public static boolean get_HighContrast();
public static int get_HorizontalScrollBarArrowWidth();
public static int get_HorizontalScrollBarHeight();
public static int get_HorizontalScrollBarThumbWidth();
public static System.Drawing.Size get_IconSize();
public static System.Drawing.Size get_IconSpacingSize();
public static int get_KanjiWindowHeight();
public static System.Drawing.Size get_MaxWindowTrackSize();
public static System.Drawing.Size get_MenuButtonSize();
public static System.Drawing.Size get_MenuCheckSize();
public static System.Drawing.Font get_MenuFont();
public static int get_MenuHeight();
public static boolean get_MidEastEnabled();
public static System.Drawing.Size get_MinimizedWindowSize();
public static System.Drawing.Size get_MinimizedWindowSpacingSize();
public static System.Drawing.Size get_MinimumWindowSize();
public static System.Drawing.Size get_MinWindowTrackSize();
public static int get_MonitorCount();
public static boolean get_MonitorsSameDisplayFormat();
public static int get_MouseButtons();
public static boolean get_MouseButtonsSwapped();
public static boolean get_MousePresent();
public static boolean get_MouseWheelPresent();
public static int get_MouseWheelScrollLines();
public static boolean get_NativeMouseWheelSupport();
public static boolean get_Network();
public static boolean get_PenWindows();
public static System.Drawing.Size get_PrimaryMonitorMaximizedWindowSize();
public static System.Drawing.Size get_PrimaryMonitorSize();
public static boolean get_RightAlignedMenus();
public static boolean get_Secure();
public static boolean get_ShowSounds();
public static System.Drawing.Size get_SmallIconSize();
public static System.Drawing.Size get_ToolWindowCaptionButtonSize();
public static int get_ToolWindowCaptionHeight();
```

```
public static System.String get_UserDomainName();
public static boolean get_UserInteractive();
public static System.String get_UserName();
public static int get_VerticalScrollBarArrowHeight();
public static int get_VerticalScrollBarThumbHeight();
public static int get_VerticalScrollBarWidth();
public static System.Drawing.Rectangle get_VirtualScreen();
public static System.Drawing.Rectangle get_WorkingArea();
```

TabAlignment Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TabAlignment
```

Members

Bottom

Left

Right

Top

TabAppearance Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TabAppearance
```

Members

Buttons

FlatButtons

Normal

TabControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "TabPages") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.TabControlDesigner
, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "SelectedIndexChanged") */
public class TabControl
   extends System.Windows.Forms.Control
```

Public Constructors

```
public TabControl();
```

Public Properties

```
public System.Windows.Forms.TabAlignment get_Alignment();
public void set_Alignment(
   System.Windows.Forms.TabAlignment value
);
public System.Windows.Forms.TabAppearance get_Appearance();
public void set_Appearance(
   System.Windows.Forms.TabAppearance value
);
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Drawing.Rectangle get_DisplayRectangle();
public System.Windows.Forms.TabDrawMode get_DrawMode();
public void set_DrawMode(
   System.Windows.Forms.TabDrawMode value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public boolean get_HotTrack();
public void set_HotTrack(
   boolean value
);
public System.Windows.Forms.ImageList get_ImageList();
public void set_ImageList(
   System.Windows.Forms.ImageList value
);
public System.Drawing.Size get_ItemSize();
public void set_ItemSize(
   System.Drawing.Size value
);
public boolean get_Multiline();
public void set_Multiline(
   boolean value
);
```

```
public System.Drawing.Point get_Padding();
public void set_Padding(
   System.Drawing.Point value
);
public int get_RowCount();
public int get_SelectedIndex();
public void set_SelectedIndex(
   int value
);
public System.Windows.Forms.TabPage get_SelectedTab();
public void set_SelectedTab(
   System.Windows.Forms.TabPage value
);
public boolean get_ShowToolTips();
public void set_ShowToolTips(
   boolean value
);
public System.Windows.Forms.TabSizeMode get_SizeMode();
public void set_SizeMode(
   System.Windows.Forms.TabSizeMode value
);
public int get_TabCount();
public System.Windows.Forms.TabControl+TabPageCollection get_TabPages();
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Methods

```
public System.Windows.Forms.Control GetControl(
   int index
);
public System.Drawing.Rectangle GetTabRect(
   int index
);
public System.String ToString();
```

Public Events

```
public void add_DrawItem(
   System.Windows.Forms.DrawItemEventHandler value
);
public void remove_DrawItem(
   System.Windows.Forms.DrawItemEventHandler value
);
public void add_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void add_SelectedIndexChanged(
   System.EventHandler value
);
public void remove_SelectedIndexChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
```

```
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected System.Windows.Forms.Control+ControlCollection CreateControlsInstance();
protected void CreateHandle();
protected System.Object[] GetItems();
protected System.Object[] GetItems(
   System.Type baseType
);
protected System.String GetToolTipText(
   System.Object item
);
protected boolean IsInputKey(
   System.Windows.Forms.Keys keyData
);
protected void OnDrawItem(
   System.Windows.Forms.DrawItemEventArgs e
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnHandleDestroyed(
   System.EventArgs e
);
protected void OnKeyDown(
   System.Windows.Forms.KeyEventArgs ke
);
protected void OnResize(
   System.EventArgs e
);
protected void OnSelectedIndexChanged(
   System.EventArgs e
);
protected void OnStyleChanged(
   System.EventArgs e
);
protected boolean ProcessKeyPreview(
   System.Windows.Forms.Message& m
);
protected void RemoveAll();
protected void UpdateTabSelection(
   boolean uiselected
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

TabControl.ControlCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ControlCollection
   extends System.Windows.Forms.Control+ControlCollection
```

Public Constructors

```
public ControlCollection(
   System.Windows.Forms.TabControl owner
);
```

Public Methods

```
public void Add(
   System.Windows.Forms.Control value
);
public void Remove(
   System.Windows.Forms.Control value
);
```

TabControl.TabPageCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class TabPageCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public TabPageCollection(
   System.Windows.Forms.TabControl owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public System.Windows.Forms.TabPage get_Item(
   int index
);
public void set_Item(
   int index,
   System.Windows.Forms.TabPage value
);
```

Public Methods

```
public void Add(
   System.Windows.Forms.TabPage value
);
public void AddRange(
   System.Windows.Forms.TabPage[] pages
);
public void Clear();
public boolean Contains(
   System.Windows.Forms.TabPage page
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Windows.Forms.TabPage page
);
public void Remove(
   System.Windows.Forms.TabPage value
);
public final void RemoveAt(
   int index
);
```

TabDrawMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TabDrawMode
```

Members

Normal

OwnerDrawFixed

TabPage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.TabPageDesigner, S
ystem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Click") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
public class TabPage
   extends System.Windows.Forms.Panel
```

Public Constructors

```
public TabPage();
public TabPage(
   System.String text
);
```

Public Properties

```
public System.Windows.Forms.AnchorStyles get_Anchor();
public void set_Anchor(
   System.Windows.Forms.AnchorStyles value
);
public System.Windows.Forms.DockStyle get_Dock();
public void set_Dock(
   System.Windows.Forms.DockStyle value
);
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public int get_ImageIndex();
public void set_ImageIndex(
   int value
);
public int get_TabIndex();
public void set_TabIndex(
   int value
);
public boolean get_TabStop();
public void set_TabStop(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.String get_ToolTipText();
public void set_ToolTipText(
   System.String value
);
public boolean get_Visible();
public void set_Visible(
   boolean value
);
```

Public Methods

```
public static System.Windows.Forms.TabPage GetTabPageOfComponent(
   System.Object comp
);
public System.String ToString();
```

Protected Methods

```
protected System.Windows.Forms.Control+ControlCollection CreateControlsInstance();
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
```

TabPage.TabPageControlCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TabPageControlCollection
   extends System.Windows.Forms.Control+ControlCollection
```

Public Constructors

```
public TabPageControlCollection(
   System.Windows.Forms.TabPage owner
);
```

Public Methods

```
public void Add(
   System.Windows.Forms.Control value
);
```

TabSizeMode Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TabSizeMode
```

Members

FillToRight

Fixed

Normal

TextBox Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TextBox
   extends System.Windows.Forms.TextBoxBase
```

Public Constructors

```
public TextBox();
```

Public Properties

```
public boolean get_AcceptsReturn();
public void set_AcceptsReturn(
   boolean value
);
public System.Windows.Forms.CharacterCasing get_CharacterCasing();
public void set_CharacterCasing(
   System.Windows.Forms.CharacterCasing value
);
public char get_PasswordChar();
public void set_PasswordChar(
   char value
);
public System.Windows.Forms.ScrollBars get_ScrollBars();
public void set_ScrollBars(
   System.Windows.Forms.ScrollBars value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.Windows.Forms.HorizontalAlignment get_TextAlign();
public void set_TextAlign(
   System.Windows.Forms.HorizontalAlignment value
);
```

Public Events

```
public void add_TextAlignChanged(
   System.EventHandler value
);
public void remove_TextAlignChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Windows.Forms.ImeMode get_DefaultImeMode();
```

Protected Methods

```
protected boolean IsInputKey(
   System.Windows.Forms.Keys keyData
);
protected void OnGotFocus(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnMouseUp(
   System.Windows.Forms.MouseEventArgs mevent
);
protected void OnTextAlignChanged(
   System.EventArgs e
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

TextBoxBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "TextChanged") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.TextBoxBaseDesigne
r, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public abstract class TextBoxBase
   extends System.Windows.Forms.Control
```

Public Properties

```
public boolean get_AcceptsTab();
public void set_AcceptsTab(
   boolean value
);
public boolean get_AutoSize();
public void set_AutoSize(
   boolean value
);
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Windows.Forms.BorderStyle value
);
public boolean get_CanUndo();
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public boolean get_HideSelection();
public void set_HideSelection(
   boolean value
);
public System.String[] get_Lines();
public void set_Lines(
   System.String[] value
);
public int get_MaxLength();
public void set_MaxLength(
   int value
);
public boolean get_Modified();
public void set_Modified(
   boolean value
);
public boolean get_Multiline();
public void set_Multiline(
   boolean value
);
public int get_PreferredHeight();
public boolean get_ReadOnly();
public void set_ReadOnly(
```

```
   boolean value
);
public System.String get_SelectedText();
public void set_SelectedText(
   System.String value
);
public int get_SelectionLength();
public void set_SelectionLength(
   int value
);
public int get_SelectionStart();
public void set_SelectionStart(
   int value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public int get_TextLength();
public boolean get_WordWrap();
public void set_WordWrap(
   boolean value
);
```

Public Methods

```
public void AppendText(
   System.String text
);
public void Clear();
public void ClearUndo();
public void Copy();
public void Cut();
public void Paste();
public void ScrollToCaret();
public void Select(
   int start,
   int length
);
public void SelectAll();
public System.String ToString();
public void Undo();
```

Public Events

```
public void add_AcceptsTabChanged(
   System.EventHandler value
);
public void remove_AcceptsTabChanged(
   System.EventHandler value
);
public void add_AutoSizeChanged(
   System.EventHandler value
);
public void remove_AutoSizeChanged(
   System.EventHandler value
);
public void add_BorderStyleChanged(
   System.EventHandler value
);
public void remove_BorderStyleChanged(
   System.EventHandler value
);
public void add_Click(
```

```
   System.EventHandler value
);
public void remove_Click(
   System.EventHandler value
);
public void add_HideSelectionChanged(
   System.EventHandler value
);
public void remove_HideSelectionChanged(
   System.EventHandler value
);
public void add_ModifiedChanged(
   System.EventHandler value
);
public void remove_ModifiedChanged(
   System.EventHandler value
);
public void add_MultilineChanged(
   System.EventHandler value
);
public void remove_MultilineChanged(
   System.EventHandler value
);
public void add_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void add_ReadOnlyChanged(
   System.EventHandler value
);
public void remove_ReadOnlyChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void CreateHandle();
protected boolean IsInputKey(
   System.Windows.Forms.Keys keyData
);
protected void OnAcceptsTabChanged(
   System.EventArgs e
);
protected void OnAutoSizeChanged(
   System.EventArgs e
);
protected void OnBorderStyleChanged(
   System.EventArgs e
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnHandleDestroyed(
   System.EventArgs e
```

```
);
protected void OnHideSelectionChanged(
   System.EventArgs e
);
protected void OnModifiedChanged(
   System.EventArgs e
);
protected void OnMultilineChanged(
   System.EventArgs e
);
protected void OnReadOnlyChanged(
   System.EventArgs e
);
protected boolean ProcessDialogKey(
   System.Windows.Forms.Keys keyData
);
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

ThreadExceptionDialog Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ThreadExceptionDialog
   extends System.Windows.Forms.Form
```

Public Constructors

```
public ThreadExceptionDialog(
   System.Exception t
);
```

TickStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TickStyle
```

Members

Both

BottomRight

None

TopLeft

Timer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Interval") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Tick") */
/** @attribute System.ComponentModel.ToolboxItemFilterAttribute(FilterString = "System.Window
s.Forms", FilterType = Allow) */
public class Timer
   extends System.ComponentModel.Component
```

Public Constructors

```
public Timer();
public Timer(
   System.ComponentModel.IContainer container
);
```

Public Properties

```
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public int get_Interval();
public void set_Interval(
   int value
);
```

Public Methods

```
public void Start();
public void Stop();
public System.String ToString();
```

Public Events

```
public void add_Tick(
   System.EventHandler value
);
public void remove_Tick(
   System.EventHandler value
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void OnTick(
   System.EventArgs e
);
```

ToolBar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "ButtonClick") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.ToolBarDesigner, S
ystem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Buttons") */
public class ToolBar
   extends System.Windows.Forms.Control
```

Public Constructors

```
public ToolBar();
```

Public Properties

```
public System.Windows.Forms.ToolBarAppearance get_Appearance();
public void set_Appearance(
   System.Windows.Forms.ToolBarAppearance value
);
public boolean get_AutoSize();
public void set_AutoSize(
   boolean value
);
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Windows.Forms.BorderStyle value
);
public System.Windows.Forms.ToolBar+ToolBarButtonCollection get_Buttons();
public System.Drawing.Size get_ButtonSize();
public void set_ButtonSize(
   System.Drawing.Size value
);
public boolean get_Divider();
public void set_Divider(
   boolean value
);
public System.Windows.Forms.DockStyle get_Dock();
public void set_Dock(
   System.Windows.Forms.DockStyle value
);
public boolean get_DropDownArrows();
public void set_DropDownArrows(
   boolean value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
```

```
public System.Windows.Forms.ImageList get_ImageList();
public void set_ImageList(
   System.Windows.Forms.ImageList value
);
public System.Drawing.Size get_ImageSize();
public System.Windows.Forms.ImeMode get_ImeMode();
public void set_ImeMode(
   System.Windows.Forms.ImeMode value
);
public System.Windows.Forms.RightToLeft get_RightToLeft();
public void set_RightToLeft(
   System.Windows.Forms.RightToLeft value
);
public boolean get_ShowToolTips();
public void set_ShowToolTips(
   boolean value
);
public boolean get_TabStop();
public void set_TabStop(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.Windows.Forms.ToolBarTextAlign get_TextAlign();
public void set_TextAlign(
   System.Windows.Forms.ToolBarTextAlign value
);
public boolean get_Wrappable();
public void set_Wrappable(
   boolean value
);
```

Public Methods

```
public System.String ToString();
```

Public Events

```
public void add_ButtonClick(
   System.Windows.Forms.ToolBarButtonClickEventHandler value
);
public void remove_ButtonClick(
   System.Windows.Forms.ToolBarButtonClickEventHandler value
);
public void add_ButtonDropDown(
   System.Windows.Forms.ToolBarButtonClickEventHandler value
);
public void remove_ButtonDropDown(
   System.Windows.Forms.ToolBarButtonClickEventHandler value
);
public void add_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
```

```
protected System.Windows.Forms.ImeMode get_DefaultImeMode();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void CreateHandle();
protected void Dispose(
   boolean disposing
);
protected void OnButtonClick(
   System.Windows.Forms.ToolBarButtonClickEventArgs e
);
protected void OnButtonDropDown(
   System.Windows.Forms.ToolBarButtonClickEventArgs e
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnResize(
   System.EventArgs e
);
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

ToolBar.ToolBarButtonCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class ToolBarButtonCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Constructors

```
public ToolBarButtonCollection(
   System.Windows.Forms.ToolBar owner
);
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public System.Windows.Forms.ToolBarButton get_Item(
   int index
);
public void set_Item(
   int index,
   System.Windows.Forms.ToolBarButton value
);
```

Public Methods

```
public int Add(
   System.String text
);
public int Add(
   System.Windows.Forms.ToolBarButton button
);
public void AddRange(
   System.Windows.Forms.ToolBarButton[] buttons
);
public final void Clear();
public boolean Contains(
   System.Windows.Forms.ToolBarButton button
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Windows.Forms.ToolBarButton button
);
public void Insert(
   int index,
   System.Windows.Forms.ToolBarButton button
);
public void Remove(
   System.Windows.Forms.ToolBarButton button
);
public final void RemoveAt(
   int index
);
```

ToolBarAppearance Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ToolBarAppearance
```

Members

Flat

Normal

ToolBarButton Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Text") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.ToolBarButtonDesig
ner, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class ToolBarButton
   extends System.ComponentModel.Component
```

Public Constructors

```
public ToolBarButton();
public ToolBarButton(
   System.String text
);
```

Public Properties

```
public System.Windows.Forms.Menu get_DropDownMenu();
public void set_DropDownMenu(
   System.Windows.Forms.Menu value
);
public boolean get_Enabled();
public void set_Enabled(
   boolean value
);
public int get_ImageIndex();
public void set_ImageIndex(
   int value
);
public System.Windows.Forms.ToolBar get_Parent();
public boolean get_PartialPush();
public void set_PartialPush(
   boolean value
);
public boolean get_Pushed();
public void set_Pushed(
   boolean value
);
public System.Drawing.Rectangle get_Rectangle();
public System.Windows.Forms.ToolBarButtonStyle get_Style();
public void set_Style(
   System.Windows.Forms.ToolBarButtonStyle value
);
public System.Object get_Tag();
public void set_Tag(
   System.Object value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.String get_ToolTipText();
public void set_ToolTipText(
   System.String value
```

```
);
public boolean get_Visible();
public void set_Visible(
   boolean value
);
```

Public Methods

```
public System.String ToString();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
```

ToolBarButtonClickEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ToolBarButtonClickEventArgs
   extends System.EventArgs
```

Public Constructors

```
public ToolBarButtonClickEventArgs(
   System.Windows.Forms.ToolBarButton button
);
```

Public Properties

```
public System.Windows.Forms.ToolBarButton get_Button();
public void set_Button(
   System.Windows.Forms.ToolBarButton value
);
```

ToolBarButtonClickEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ToolBarButtonClickEventHandler(
   System.Object sender,
   System.Windows.Forms.ToolBarButtonClickEventArgs e
);
```

ToolBarButtonStyle Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ToolBarButtonStyle
```

Members

DropDownButton

PushButton

Separator

ToggleButton

ToolBarTextAlign Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ToolBarTextAlign
```

Members

Right

Underneath

ToolTip Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemFilterAttribute(FilterString = "System.Window
s.Forms", FilterType = Allow) */
/** @attribute System.ComponentModel.ProvidePropertyAttribute(PropertyName = "ToolTip", Recei
verTypeName = "System.Windows.Forms.Control, System.Windows.Forms, Version=1.0.3300.0, Cultur
e=neutral, PublicKeyToken=b77a5c561934e089") */
public class ToolTip
   extends System.ComponentModel.Component
   implements System.ComponentModel.IExtenderProvider
```

Public Constructors

```
public ToolTip();
public ToolTip(
   System.ComponentModel.IContainer cont
);
```

Public Properties

```
public boolean get_Active();
public void set_Active(
   boolean value
);
public int get_AutomaticDelay();
public void set_AutomaticDelay(
   int value
);
public int get_AutoPopDelay();
public void set_AutoPopDelay(
   int value
);
public int get_InitialDelay();
public void set_InitialDelay(
   int value
);
public int get_ReshowDelay();
public void set_ReshowDelay(
   int value
);
public boolean get_ShowAlways();
public void set_ShowAlways(
   boolean value
);
```

Public Methods

```
public final boolean CanExtend(
   System.Object target
);
/** @attribute System.ComponentModel.DefaultValueAttribute(Value = ) */
/** @attribute System.Windows.Forms.SRDescriptionAttribute(Description = "Determines the tool
tip shown when the mouse hovers over the control.") */
/** @attribute System.ComponentModel.LocalizableAttribute(IsLocalizable = true) */
public System.String GetToolTip(
```

```
   System.Windows.Forms.Control control
);
public void RemoveAll();
public void SetToolTip(
   System.Windows.Forms.Control control,
   System.String caption
);
public System.String ToString();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

TrackBar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.TrackBarDesigner,
System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Value") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Scroll") */
public class TrackBar
   extends System.Windows.Forms.Control
   implements System.ComponentModel.ISupportInitialize
```

Public Constructors

```
public TrackBar();
```

Public Properties

```
public boolean get_AutoSize();
public void set_AutoSize(
   boolean value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Drawing.Font get_Font();
public void set_Font(
   System.Drawing.Font value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.Windows.Forms.ImeMode get_ImeMode();
public void set_ImeMode(
   System.Windows.Forms.ImeMode value
);
public int get_LargeChange();
public void set_LargeChange(
   int value
);
public int get_Maximum();
public void set_Maximum(
   int value
);
public int get_Minimum();
public void set_Minimum(
   int value
);
public System.Windows.Forms.Orientation get_Orientation();
public void set_Orientation(
   System.Windows.Forms.Orientation value
);
public int get_SmallChange();
public void set_SmallChange(
   int value
);
```

```
public System.String get_Text();
public void set_Text(
   System.String value
);
public int get_TickFrequency();
public void set_TickFrequency(
   int value
);
public System.Windows.Forms.TickStyle get_TickStyle();
public void set_TickStyle(
   System.Windows.Forms.TickStyle value
);
public int get_Value();
public void set_Value(
   int value
);
```

Public Methods

```
public final void BeginInit();
public final void EndInit();
public void SetRange(
   int minValue,
   int maxValue
);
public System.String ToString();
```

Public Events

```
public void add_Click(
   System.EventHandler value
);
public void remove_Click(
   System.EventHandler value
);
public void add_DoubleClick(
   System.EventHandler value
);
public void remove_DoubleClick(
   System.EventHandler value
);
public void add_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void add_Scroll(
   System.EventHandler value
);
public void remove_Scroll(
   System.EventHandler value
);
public void add_ValueChanged(
   System.EventHandler value
);
public void remove_ValueChanged(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Windows.Forms.ImeMode get_DefaultImeMode();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void CreateHandle();
protected boolean IsInputKey(
   System.Windows.Forms.Keys keyData
);
protected void OnBackColorChanged(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnScroll(
   System.EventArgs e
);
protected void OnValueChanged(
   System.EventArgs e
);
protected void SetBoundsCore(
   int x,
   int y,
   int width,
   int height,
   System.Windows.Forms.BoundsSpecified specified
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

TreeNode Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.ComponentModel.TypeConverterAttribute(ConverterTypeName = "System.Windo
ws.Forms.TreeNodeConverter, System.Windows.Forms, Version=1.0.3300.0, Culture=neutral, Public
KeyToken=b77a5c561934e089") */
public class TreeNode
   extends System.MarshalByRefObject
   implements System.ICloneable, System.Runtime.Serialization.ISerializable
```

Public Constructors

```
public TreeNode();
public TreeNode(
   System.String text
);
public TreeNode(
   System.String text,
   System.Windows.Forms.TreeNode[] children
);
public TreeNode(
   System.String text,
   int imageIndex,
   int selectedImageIndex
);
public TreeNode(
   System.String text,
   int imageIndex,
   int selectedImageIndex,
   System.Windows.Forms.TreeNode[] children
);
```

Public Properties

```
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Rectangle get_Bounds();
public boolean get_Checked();
public void set_Checked(
   boolean value
);
public System.Windows.Forms.TreeNode get_FirstNode();
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public System.String get_FullPath();
public System.IntPtr get_Handle();
public int get_ImageIndex();
public void set_ImageIndex(
   int value
);
public int get_Index();
public boolean get_IsEditing();
public boolean get_IsExpanded();
```

```
public boolean get_IsSelected();
public boolean get_IsVisible();
public System.Windows.Forms.TreeNode get_LastNode();
public System.Windows.Forms.TreeNode get_NextNode();
public System.Windows.Forms.TreeNode get_NextVisibleNode();
public System.Drawing.Font get_NodeFont();
public void set_NodeFont(
   System.Drawing.Font value
);
public System.Windows.Forms.TreeNodeCollection get_Nodes();
public System.Windows.Forms.TreeNode get_Parent();
public System.Windows.Forms.TreeNode get_PrevNode();
public System.Windows.Forms.TreeNode get_PrevVisibleNode();
public int get_SelectedImageIndex();
public void set_SelectedImageIndex(
   int value
);
public System.Object get_Tag();
public void set_Tag(
   System.Object value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.Windows.Forms.TreeView get_TreeView();
```

Public Methods

```
public void BeginEdit();
public System.Object Clone();
public void Collapse();
public void EndEdit(
   boolean cancel
);
public void EnsureVisible();
public void Expand();
public void ExpandAll();
public static System.Windows.Forms.TreeNode FromHandle(
   System.Windows.Forms.TreeView tree,
   System.IntPtr handle
);
public int GetNodeCount(
   boolean includeSubTrees
);
public void Remove();
public void Toggle();
public System.String ToString();
```

TreeNodeCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
/** @attribute System.ComponentModel.EditorAttribute(EditorBaseTypeName = "System.Drawing.Des
ign.UITypeEditor, System.Drawing, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7
f11d50a3a", EditorTypeName = "System.Windows.Forms.Design.TreeNodeCollectionEditor, System.De
sign, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class TreeNodeCollection
   extends System.Object
   implements System.Collections.IList, System.Collections.ICollection, System.Collections.IE
numerable
```

Public Properties

```
public final int get_Count();
public final boolean get_IsReadOnly();
public System.Windows.Forms.TreeNode get_Item(
   int index
);
public void set_Item(
   int index,
   System.Windows.Forms.TreeNode value
);
```

Public Methods

```
public int Add(
   System.Windows.Forms.TreeNode node
);
public System.Windows.Forms.TreeNode Add(
   System.String text
);
public void AddRange(
   System.Windows.Forms.TreeNode[] nodes
);
public void Clear();
public boolean Contains(
   System.Windows.Forms.TreeNode node
);
public final void CopyTo(
   System.Array dest,
   int index
);
public final System.Collections.IEnumerator GetEnumerator();
public int IndexOf(
   System.Windows.Forms.TreeNode node
);
public void Insert(
   int index,
   System.Windows.Forms.TreeNode node
);
public void Remove(
   System.Windows.Forms.TreeNode node
);
public void RemoveAt(
   int index
);
```

TreeNodeConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TreeNodeConverter
   extends System.ComponentModel.TypeConverter
```

Public Constructors

```
public TreeNodeConverter();
```

Public Methods

```
public boolean CanConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Type destinationType
);
public System.Object ConvertTo(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Globalization.CultureInfo culture,
   System.Object value,
   System.Type destinationType
);
```

TreeView Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "AfterSelect") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.TreeViewDesigner,
System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DefaultPropertyAttribute(Name = "Nodes") */
public class TreeView
   extends System.Windows.Forms.Control
```

Public Constructors

```
public TreeView();
```

Public Properties

```
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Windows.Forms.BorderStyle value
);
public boolean get_CheckBoxes();
public void set_CheckBoxes(
   boolean value
);
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public boolean get_FullRowSelect();
public void set_FullRowSelect(
   boolean value
);
public boolean get_HideSelection();
public void set_HideSelection(
   boolean value
);
public boolean get_HotTracking();
public void set_HotTracking(
   boolean value
);
public int get_ImageIndex();
public void set_ImageIndex(
   int value
);
public System.Windows.Forms.ImageList get_ImageList();
public void set_ImageList(
   System.Windows.Forms.ImageList value
);
public int get_Indent();
```

```
public void set_Indent(
   int value
);
public int get_ItemHeight();
public void set_ItemHeight(
   int value
);
public boolean get_LabelEdit();
public void set_LabelEdit(
   boolean value
);
public System.Windows.Forms.TreeNodeCollection get_Nodes();
public System.String get_PathSeparator();
public void set_PathSeparator(
   System.String value
);
public boolean get_Scrollable();
public void set_Scrollable(
   boolean value
);
public int get_SelectedImageIndex();
public void set_SelectedImageIndex(
   int value
);
public System.Windows.Forms.TreeNode get_SelectedNode();
public void set_SelectedNode(
   System.Windows.Forms.TreeNode value
);
public boolean get_ShowLines();
public void set_ShowLines(
   boolean value
);
public boolean get_ShowPlusMinus();
public void set_ShowPlusMinus(
   boolean value
);
public boolean get_ShowRootLines();
public void set_ShowRootLines(
   boolean value
);
public boolean get_Sorted();
public void set_Sorted(
   boolean value
);
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.Windows.Forms.TreeNode get_TopNode();
public int get_VisibleCount();
```

Public Methods

```
public void BeginUpdate();
public void CollapseAll();
public void EndUpdate();
public void ExpandAll();
public System.Windows.Forms.TreeNode GetNodeAt(
   System.Drawing.Point pt
);
public System.Windows.Forms.TreeNode GetNodeAt(
   int x,
   int y
);
public int GetNodeCount(
   boolean includeSubTrees
```

```
);
public System.String ToString();
```

Public Events

```
public void add_AfterCheck(
   System.Windows.Forms.TreeViewEventHandler value
);
public void remove_AfterCheck(
   System.Windows.Forms.TreeViewEventHandler value
);
public void add_AfterCollapse(
   System.Windows.Forms.TreeViewEventHandler value
);
public void remove_AfterCollapse(
   System.Windows.Forms.TreeViewEventHandler value
);
public void add_AfterExpand(
   System.Windows.Forms.TreeViewEventHandler value
);
public void remove_AfterExpand(
   System.Windows.Forms.TreeViewEventHandler value
);
public void add_AfterLabelEdit(
   System.Windows.Forms.NodeLabelEditEventHandler value
);
public void remove_AfterLabelEdit(
   System.Windows.Forms.NodeLabelEditEventHandler value
);
public void add_AfterSelect(
   System.Windows.Forms.TreeViewEventHandler value
);
public void remove_AfterSelect(
   System.Windows.Forms.TreeViewEventHandler value
);
public void add_BeforeCheck(
   System.Windows.Forms.TreeViewCancelEventHandler value
);
public void remove_BeforeCheck(
   System.Windows.Forms.TreeViewCancelEventHandler value
);
public void add_BeforeCollapse(
   System.Windows.Forms.TreeViewCancelEventHandler value
);
public void remove_BeforeCollapse(
   System.Windows.Forms.TreeViewCancelEventHandler value
);
public void add_BeforeExpand(
   System.Windows.Forms.TreeViewCancelEventHandler value
);
public void remove_BeforeExpand(
   System.Windows.Forms.TreeViewCancelEventHandler value
);
public void add_BeforeLabelEdit(
   System.Windows.Forms.NodeLabelEditEventHandler value
);
public void remove_BeforeLabelEdit(
   System.Windows.Forms.NodeLabelEditEventHandler value
);
public void add_BeforeSelect(
   System.Windows.Forms.TreeViewCancelEventHandler value
);
public void remove_BeforeSelect(
   System.Windows.Forms.TreeViewCancelEventHandler value
);
public void add_ItemDrag(
```

```
   System.Windows.Forms.ItemDragEventHandler value
);
public void remove_ItemDrag(
   System.Windows.Forms.ItemDragEventHandler value
);
public void add_Paint(
   System.Windows.Forms.PaintEventHandler value
);
public void remove_Paint(
   System.Windows.Forms.PaintEventHandler value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
protected void CreateHandle();
protected void Dispose(
   boolean disposing
);
protected System.Windows.Forms.OwnerDrawPropertyBag GetItemRenderStyles(
   System.Windows.Forms.TreeNode node,
   int state
);
protected boolean IsInputKey(
   System.Windows.Forms.Keys keyData
);
protected void OnAfterCheck(
   System.Windows.Forms.TreeViewEventArgs e
);
protected void OnAfterCollapse(
   System.Windows.Forms.TreeViewEventArgs e
);
protected void OnAfterExpand(
   System.Windows.Forms.TreeViewEventArgs e
);
protected void OnAfterLabelEdit(
   System.Windows.Forms.NodeLabelEditEventArgs e
);
protected void OnAfterSelect(
   System.Windows.Forms.TreeViewEventArgs e
);
protected void OnBeforeCheck(
   System.Windows.Forms.TreeViewCancelEventArgs e
);
protected void OnBeforeCollapse(
   System.Windows.Forms.TreeViewCancelEventArgs e
);
protected void OnBeforeExpand(
   System.Windows.Forms.TreeViewCancelEventArgs e
);
protected void OnBeforeLabelEdit(
   System.Windows.Forms.NodeLabelEditEventArgs e
);
protected void OnBeforeSelect(
   System.Windows.Forms.TreeViewCancelEventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnHandleDestroyed(
```

```
    System.EventArgs e
);
protected void OnItemDrag(
    System.Windows.Forms.ItemDragEventArgs e
);
protected void OnKeyDown(
    System.Windows.Forms.KeyEventArgs e
);
protected void OnKeyPress(
    System.Windows.Forms.KeyPressEventArgs e
);
protected void OnKeyUp(
    System.Windows.Forms.KeyEventArgs e
);
protected void WndProc(
    System.Windows.Forms.Message& m
);
```

TreeViewAction Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum TreeViewAction
```

Members

ByKeyboard

ByMouse

Collapse

Expand

Unknown

TreeViewCancelEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TreeViewCancelEventArgs
   extends System.ComponentModel.CancelEventArgs
```

Public Constructors

```
public TreeViewCancelEventArgs(
   System.Windows.Forms.TreeNode node,
   boolean cancel,
   System.Windows.Forms.TreeViewAction action
);
```

Public Properties

```
public System.Windows.Forms.TreeViewAction get_Action();
public System.Windows.Forms.TreeNode get_Node();
```

TreeViewCancelEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void TreeViewCancelEventHandler(
   System.Object sender,
   System.Windows.Forms.TreeViewCancelEventArgs e
);
```

TreeViewEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TreeViewEventArgs
   extends System.EventArgs
```

Public Constructors

```
public TreeViewEventArgs(
   System.Windows.Forms.TreeNode node
);
public TreeViewEventArgs(
   System.Windows.Forms.TreeNode node,
   System.Windows.Forms.TreeViewAction action
);
```

Public Properties

```
public System.Windows.Forms.TreeViewAction get_Action();
public System.Windows.Forms.TreeNode get_Node();
```

TreeViewEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void TreeViewEventHandler(
   System.Object sender,
   System.Windows.Forms.TreeViewEventArgs e
);
```

TreeViewImageIndexConverter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class TreeViewImageIndexConverter
   extends System.Windows.Forms.ImageIndexConverter
```

Public Constructors

```
public TreeViewImageIndexConverter();
```

Protected Properties

```
protected boolean get_IncludeNoneAsStandardValue();
```

UICues Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum UICues
```

Members

Changed

ChangeFocus

ChangeKeyboard

None

ShowFocus

ShowKeyboard

Shown

UICuesEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UICuesEventArgs
   extends System.EventArgs
```

Public Constructors

```
public UICuesEventArgs(
   System.Windows.Forms.UICues uicues
);
```

Public Properties

```
public System.Windows.Forms.UICues get_Changed();
public boolean get_ChangeFocus();
public boolean get_ChangeKeyboard();
public boolean get_ShowFocus();
public boolean get_ShowKeyboard();
```

UICuesEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void UICuesEventHandler(
   System.Object sender,
   System.Windows.Forms.UICuesEventArgs e
);
```

UpDownBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.UpDownBaseDesigner
, System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public abstract class UpDownBase
   extends System.Windows.Forms.ContainerControl
```

Public Constructors

```
public UpDownBase();
```

Public Properties

```
public boolean get_AutoScroll();
public void set_AutoScroll(
   boolean value
);
public System.Drawing.Size get_AutoScrollMargin();
public void set_AutoScrollMargin(
   System.Drawing.Size value
);
public System.Drawing.Size get_AutoScrollMinSize();
public void set_AutoScrollMinSize(
   System.Drawing.Size value
);
public System.Drawing.Color get_BackColor();
public void set_BackColor(
   System.Drawing.Color value
);
public System.Drawing.Image get_BackgroundImage();
public void set_BackgroundImage(
   System.Drawing.Image value
);
public System.Windows.Forms.BorderStyle get_BorderStyle();
public void set_BorderStyle(
   System.Windows.Forms.BorderStyle value
);
public System.Windows.Forms.ContextMenu get_ContextMenu();
public void set_ContextMenu(
   System.Windows.Forms.ContextMenu value
);
public System.Windows.Forms.ScrollableControl+DockPaddingEdges get_DockPadding();
public boolean get_Focused();
public System.Drawing.Color get_ForeColor();
public void set_ForeColor(
   System.Drawing.Color value
);
public boolean get_InterceptArrowKeys();
public void set_InterceptArrowKeys(
   boolean value
);
public int get_PreferredHeight();
public boolean get_ReadOnly();
public void set_ReadOnly(
   boolean value
);
```

```
public System.String get_Text();
public void set_Text(
   System.String value
);
public System.Windows.Forms.HorizontalAlignment get_TextAlign();
public void set_TextAlign(
   System.Windows.Forms.HorizontalAlignment value
);
public System.Windows.Forms.LeftRightAlignment get_UpDownAlign();
public void set_UpDownAlign(
   System.Windows.Forms.LeftRightAlignment value
);
```

Public Methods

```
public abstract void DownButton();
public void Select(
   int start,
   int length
);
public abstract void UpButton();
```

Protected Properties

```
protected boolean get_ChangingText();
protected void set_ChangingText(
   boolean value
);
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
protected boolean get_UserEdit();
protected void set_UserEdit(
   boolean value
);
```

Protected Methods

```
protected void OnChanged(
   System.Object source,
   System.EventArgs e
);
protected void OnFontChanged(
   System.EventArgs e
);
protected void OnHandleCreated(
   System.EventArgs e
);
protected void OnLayout(
   System.Windows.Forms.LayoutEventArgs e
);
protected void OnMouseWheel(
   System.Windows.Forms.MouseEventArgs e
);
protected void OnTextBoxKeyDown(
   System.Object source,
   System.Windows.Forms.KeyEventArgs e
);
protected void OnTextBoxKeyPress(
   System.Object source,
   System.Windows.Forms.KeyPressEventArgs e
);
protected void OnTextBoxLostFocus(
```

```
    System.Object source,
    System.EventArgs e
);
protected void OnTextBoxResize(
    System.Object source,
    System.EventArgs e
);
protected void OnTextBoxTextChanged(
    System.Object source,
    System.EventArgs e
);
protected void SetBoundsCore(
    int x,
    int y,
    int width,
    int height,
    System.Windows.Forms.BoundsSpecified specified
);
protected abstract void UpdateEditText();
protected void ValidateEditText();
```

UpDownEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UpDownEventArgs
   extends System.EventArgs
```

Public Constructors

```
public UpDownEventArgs(
   int buttonPushed
);
```

Public Properties

```
public int get_ButtonID();
```

UpDownEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void UpDownEventHandler(
   System.Object source,
   System.Windows.Forms.UpDownEventArgs e
);
```

UserControl Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IDesigner", DesignerTypeName = "System.Windows.Forms.Design.ControlDesigner, S
ystem.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
/** @attribute System.ComponentModel.DesignerCategoryAttribute(Category = "UserControl") */
/** @attribute System.ComponentModel.DefaultEventAttribute(Name = "Load") */
/** @attribute System.ComponentModel.DesignerAttribute(DesignerBaseTypeName = "System.Compone
ntModel.Design.IRootDesigner, System, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b77
a5c561934e089", DesignerTypeName = "System.Windows.Forms.Design.UserControlDocumentDesigner,
System.Design, Version=1.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a") */
public class UserControl
   extends System.Windows.Forms.ContainerControl
```

Public Constructors

```
public UserControl();
```

Public Properties

```
public System.String get_Text();
public void set_Text(
   System.String value
);
```

Public Events

```
public void add_Load(
   System.EventHandler value
);
public void remove_Load(
   System.EventHandler value
);
```

Protected Properties

```
protected System.Drawing.Size get_DefaultSize();
```

Protected Methods

```
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnCreateControl();
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnLoad(
   System.EventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
protected void OnMouseDown(
   System.Windows.Forms.MouseEventArgs e
);
/** @attribute System.ComponentModel.EditorBrowsableAttribute(State = Advanced) */
```

```
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

View Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum View
```

Members

Details

LargeIcon

List

SmallIcon

VScrollBar Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class VScrollBar
   extends System.Windows.Forms.ScrollBar
```

Public Constructors

```
public VScrollBar();
```

Public Properties

```
public System.Windows.Forms.RightToLeft get_RightToLeft();
public void set_RightToLeft(
   System.Windows.Forms.RightToLeft value
);
```

Protected Properties

```
protected System.Windows.Forms.CreateParams get_CreateParams();
protected System.Drawing.Size get_DefaultSize();
```

System.Windows.Forms.Design Namespace (Visual J# Syntax)

Classes

AnchorEditor

AxImporter

AxImporter.Options

AxParameterData

AxWrapperGen

ComponentDocumentDesigner

ComponentEditorForm

ComponentEditorPage

ComponentTray

ControlDesigner

ControlDesigner.ControlDesignerAccessibleObject

DockEditor

DocumentDesigner

EventHandlerService

EventsTab

FileNameEditor

FolderNameEditor

FolderNameEditor.FolderBrowser

MenuCommands

ParentControlDesigner

PropertyTab

ScrollableControlDesigner

WindowsFormsComponentEditor

Interfaces

AxImporter.IReferenceResolver

IMenuEditorService

IUIService

IWindowsFormsEditorService

Enumerations

FolderNameEditor.FolderBrowserFolder

FolderNameEditor.FolderBrowserStyles

SelectionRules

AnchorEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AnchorEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public AnchorEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

AxImporter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AxImporter
   extends System.Object
```

Public Constructors

```
public AxImporter(
   System.Windows.Forms.Design.AxImporter+Options options
);
```

Public Properties

```
public System.String[] get_GeneratedAssemblies();
public System.String[] get_GeneratedSources();
public System.Runtime.InteropServices.TYPELIBATTR[] get_GeneratedTypeLibAttributes();
```

Public Methods

```
public System.String GenerateFromFile(
   System.IO.FileInfo file
);
public System.String GenerateFromTypeLibrary(
   System.Runtime.InteropServices.UCOMITypeLib typeLib
);
public System.String GenerateFromTypeLibrary(
   System.Runtime.InteropServices.UCOMITypeLib typeLib,
   System.Guid clsid
);
public static System.String GetFileOfTypeLib(
   System.Runtime.InteropServices.TYPELIBATTR& tlibattr
);
```

AxImporter.IReferenceResolver Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IReferenceResolver
```

Public Methods

```
System.String ResolveActiveXReference(
   System.Runtime.InteropServices.UCOMITypeLib typeLib
);
System.String ResolveComReference(
   System.Reflection.AssemblyName name
);
System.String ResolveComReference(
   System.Runtime.InteropServices.UCOMITypeLib typeLib
);
System.String ResolveManagedReference(
   System.String assemName
);
```

AxImporter.Options Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class Options
   extends System.Object
```

Public Constructors

```
public Options();
```

Public Fields

```
public boolean delaySign;
public boolean genSources;
public System.String keyContainer;
public System.String keyFile;
public System.Reflection.StrongNameKeyPair keyPair;
public boolean noLogo;
public System.String outputDirectory;
public System.String outputName;
public boolean overwriteRCW;
public ubyte[] publicKey;
public System.Windows.Forms.Design.AxImporter+IReferenceResolver references;
public boolean silentMode;
public boolean verboseMode;
```

AxParameterData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AxParameterData
   extends System.Object
```

Public Constructors

```
public AxParameterData(
   System.Reflection.ParameterInfo info
);
public AxParameterData(
   System.Reflection.ParameterInfo info,
   boolean ignoreByRefs
);
public AxParameterData(
   System.String inname,
   System.String typeName
);
public AxParameterData(
   System.String inname,
   System.Type type
);
```

Public Properties

```
public System.CodeDom.FieldDirection get_Direction();
public boolean get_IsByRef();
public boolean get_IsIn();
public boolean get_IsOptional();
public boolean get_IsOut();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Type get_ParameterType();
public System.String get_TypeName();
```

Public Methods

```
public static System.Windows.Forms.Design.AxParameterData[] Convert(
   System.Reflection.ParameterInfo[] infos
);
public static System.Windows.Forms.Design.AxParameterData[] Convert(
   System.Reflection.ParameterInfo[] infos,
   boolean ignoreByRefs
);
```

AxWrapperGen Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class AxWrapperGen
   extends System.Object
```

Public Constructors

```
public AxWrapperGen(
   System.Type axType
);
```

Public Fields

```
public static System.Collections.ArrayList GeneratedSources;
```

ComponentDocumentDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ComponentDocumentDesigner
   extends System.ComponentModel.Design.ComponentDesigner
   implements System.ComponentModel.Design.IRootDesigner, System.Drawing.Design.IToolboxUser,
System.ComponentModel.Design.ITypeDescriptorFilterService
```

Public Constructors

```
public ComponentDocumentDesigner();
```

Public Properties

```
public System.Windows.Forms.Control get_Control();
public boolean get_TrayAutoArrange();
public void set_TrayAutoArrange(
   boolean value
);
public boolean get_TrayLargeIcon();
public void set_TrayLargeIcon(
   boolean value
);
```

Public Methods

```
public void Initialize(
   System.ComponentModel.IComponent component
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected boolean GetToolSupported(
   System.Drawing.Design.ToolboxItem tool
);
protected void PreFilterProperties(
   System.Collections.IDictionary properties
);
```

ComponentEditorForm Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
public class ComponentEditorForm
   extends System.Windows.Forms.Form
```

Public Constructors

```
public ComponentEditorForm(
   System.Object component,
   System.Type[] pageTypes
);
```

Public Methods

```
public boolean PreProcessMessage(
   System.Windows.Forms.Message& msg
);
public System.Windows.Forms.DialogResult ShowForm();
public System.Windows.Forms.DialogResult ShowForm(
   int page
);
public System.Windows.Forms.DialogResult ShowForm(
   System.Windows.Forms.IWin32Window owner
);
public System.Windows.Forms.DialogResult ShowForm(
   System.Windows.Forms.IWin32Window owner,
   int page
);
```

Protected Methods

```
protected void OnActivated(
   System.EventArgs e
);
protected void OnHelpRequested(
   System.Windows.Forms.HelpEventArgs e
);
protected void OnSelChangeSelector(
   System.Object source,
   System.Windows.Forms.TreeViewEventArgs e
);
```

ComponentEditorPage Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class ComponentEditorPage
   extends System.Windows.Forms.Panel
```

Public Constructors

```
public ComponentEditorPage();
```

Public Properties

```
public boolean get_CommitOnDeactivate();
public void set_CommitOnDeactivate(
   boolean value
);
public System.Drawing.Icon get_Icon();
public void set_Icon(
   System.Drawing.Icon value
);
public System.String get_Title();
```

Public Methods

```
public void Activate();
public void ApplyChanges();
public void Deactivate();
public System.Windows.Forms.Control GetControl();
public boolean IsPageMessage(
   System.Windows.Forms.Message& msg
);
public void OnApplyComplete();
public void SetComponent(
   System.ComponentModel.IComponent component
);
public void SetSite(
   System.Windows.Forms.IComponentEditorPageSite site
);
public void ShowHelp();
public boolean SupportsHelp();
```

Protected Properties

```
protected System.ComponentModel.IComponent get_Component();
protected void set_Component(
   System.ComponentModel.IComponent value
);
protected System.Windows.Forms.CreateParams get_CreateParams();
protected boolean get_FirstActivate();
protected void set_FirstActivate(
   boolean value
);
protected int get_Loading();
protected void set_Loading(
```

```
   int value
);
protected boolean get_LoadRequired();
protected void set_LoadRequired(
   boolean value
);
protected System.Windows.Forms.IComponentEditorPageSite get_PageSite();
protected void set_PageSite(
   System.Windows.Forms.IComponentEditorPageSite value
);
```

Protected Methods

```
protected void EnterLoadingMode();
protected void ExitLoadingMode();
protected System.ComponentModel.IComponent GetSelectedComponent();
protected boolean IsFirstActivate();
protected boolean IsLoading();
protected abstract void LoadComponent();
protected void ReloadComponent();
protected abstract void SaveComponent();
protected void SetDirty();
```

ComponentTray Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.DesignTimeVisibleAttribute(Visible = false) */
/** @attribute System.ComponentModel.ToolboxItemAttribute(ToolboxItemTypeName = "") */
/** @attribute System.ComponentModel.ProvidePropertyAttribute(PropertyName = "Location", Rece
iverTypeName = "System.ComponentModel.IComponent, System, Version=1.0.3300.0, Culture=neutral
, PublicKeyToken=b77a5c561934e089") */
public class ComponentTray
   extends System.Windows.Forms.ScrollableControl
   implements System.ComponentModel.IExtenderProvider
```

Public Constructors

```
public ComponentTray(
   System.ComponentModel.Design.IDesigner mainDesigner,
   System.IServiceProvider serviceProvider
);
```

Public Properties

```
public boolean get_AutoArrange();
public void set_AutoArrange(
   boolean value
);
public int get_ComponentCount();
public boolean get_ShowLargeIcons();
public void set_ShowLargeIcons(
   boolean value
);
```

Public Methods

```
public void AddComponent(
   System.ComponentModel.IComponent component
);
public void CreateComponentFromTool(
   System.Drawing.Design.ToolboxItem tool
);
/** @attribute System.ComponentModel.DesignOnlyAttribute(IsDesignOnly = true) */
/** @attribute System.ComponentModel.BrowsableAttribute(Browsable = false) */
/** @attribute System.Design.SRDescriptionAttribute() */
/** @attribute System.ComponentModel.CategoryAttribute(Action = System.ComponentModel.Categor
yAttribute, Appearance = System.ComponentModel.CategoryAttribute, Behavior = System.Component
Model.CategoryAttribute, Data = System.ComponentModel.CategoryAttribute, Default = System.Com
ponentModel.CategoryAttribute, Design = System.ComponentModel.CategoryAttribute, DragDrop = S
ystem.ComponentModel.CategoryAttribute, Focus = System.ComponentModel.CategoryAttribute, Form
at = System.ComponentModel.CategoryAttribute, Key = System.ComponentModel.CategoryAttribute,
Layout = System.ComponentModel.CategoryAttribute, Mouse = System.ComponentModel.CategoryAttri
bute, WindowStyle = System.ComponentModel.CategoryAttribute, Category = "Layout") */
/** @attribute System.ComponentModel.LocalizableAttribute(IsLocalizable = false) */
public System.Drawing.Point GetLocation(
   System.ComponentModel.IComponent receiver
);
public void RemoveComponent(
   System.ComponentModel.IComponent component
```

```
);
public void SetLocation(
   System.ComponentModel.IComponent receiver,
   System.Drawing.Point location
);
```

Protected Methods

```
protected boolean CanCreateComponentFromTool(
   System.Drawing.Design.ToolboxItem tool
);
protected boolean CanDisplayComponent(
   System.ComponentModel.IComponent component
);
protected void DisplayError(
   System.Exception e
);
protected void Dispose(
   boolean disposing
);
protected System.Object GetService(
   System.Type serviceType
);
protected void OnDoubleClick(
   System.EventArgs e
);
protected void OnDragDrop(
   System.Windows.Forms.DragEventArgs de
);
protected void OnDragEnter(
   System.Windows.Forms.DragEventArgs de
);
protected void OnDragLeave(
   System.EventArgs e
);
protected void OnDragOver(
   System.Windows.Forms.DragEventArgs de
);
protected void OnGiveFeedback(
   System.Windows.Forms.GiveFeedbackEventArgs gfevent
);
protected void OnLayout(
   System.Windows.Forms.LayoutEventArgs levent
);
protected void OnLostCapture();
protected void OnMouseDown(
   System.Windows.Forms.MouseEventArgs e
);
protected void OnMouseMove(
   System.Windows.Forms.MouseEventArgs e
);
protected void OnMouseUp(
   System.Windows.Forms.MouseEventArgs e
);
protected void OnPaint(
   System.Windows.Forms.PaintEventArgs pe
);
protected void OnSetCursor();
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

ControlDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ControlDesigner
   extends System.ComponentModel.Design.ComponentDesigner
```

Public Constructors

```
public ControlDesigner();
```

Public Properties

```
public System.Windows.Forms.AccessibleObject get_AccessibilityObject();
public System.Collections.ICollection get_AssociatedComponents();
public System.Windows.Forms.Control get_Control();
public System.Windows.Forms.Design.SelectionRules get_SelectionRules();
```

Public Methods

```
public boolean CanBeParentedTo(
   System.ComponentModel.Design.IDesigner parentDesigner
);
public void Initialize(
   System.ComponentModel.IComponent component
);
public void InitializeNonDefault();
public void OnSetComponentDefaults();
```

Protected Properties

```
protected boolean get_EnableDragRect();
```

Protected Methods

```
protected void BaseWndProc(
   System.Windows.Forms.Message& m
);
protected void DefWndProc(
   System.Windows.Forms.Message& m
);
protected void DisplayError(
   System.Exception e
);
protected void Dispose(
   boolean disposing
);
protected void EnableDragDrop(
   boolean value
);
protected boolean GetHitTest(
   System.Drawing.Point point
);
```

```
protected void HookChildControls(
   System.Windows.Forms.Control firstChild
);
protected void OnContextMenu(
   int x,
   int y
);
protected void OnCreateHandle();
protected void OnDragDrop(
   System.Windows.Forms.DragEventArgs de
);
protected void OnDragEnter(
   System.Windows.Forms.DragEventArgs de
);
protected void OnDragLeave(
   System.EventArgs e
);
protected void OnDragOver(
   System.Windows.Forms.DragEventArgs de
);
protected void OnGiveFeedback(
   System.Windows.Forms.GiveFeedbackEventArgs e
);
protected void OnMouseDragBegin(
   int x,
   int y
);
protected void OnMouseDragEnd(
   boolean cancel
);
protected void OnMouseDragMove(
   int x,
   int y
);
protected void OnMouseEnter();
protected void OnMouseHover();
protected void OnMouseLeave();
protected void OnPaintAdornments(
   System.Windows.Forms.PaintEventArgs pe
);
protected void OnSetCursor();
protected void PreFilterProperties(
   System.Collections.IDictionary properties
);
protected void UnhookChildControls(
   System.Windows.Forms.Control firstChild
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

Protected Fields

```
protected System.Windows.Forms.AccessibleObject accessibilityObj;
protected static final System.Drawing.Point InvalidPoint;
```

ControlDesigner.ControlDesignerAccessibleObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.ComVisibleAttribute(Value = true) */
public class ControlDesignerAccessibleObject
   extends System.Windows.Forms.AccessibleObject
```

Public Constructors

```
public ControlDesignerAccessibleObject(
   System.Windows.Forms.Design.ControlDesigner designer,
   System.Windows.Forms.Control control
);
```

Public Properties

```
public System.Drawing.Rectangle get_Bounds();
public System.String get_DefaultAction();
public System.String get_Description();
public System.String get_Name();
public System.Windows.Forms.AccessibleObject get_Parent();
public System.Windows.Forms.AccessibleRole get_Role();
public System.Windows.Forms.AccessibleStates get_State();
public System.String get_Value();
```

Public Methods

```
public System.Windows.Forms.AccessibleObject GetChild(
   int index
);
public int GetChildCount();
public System.Windows.Forms.AccessibleObject GetFocused();
public System.Windows.Forms.AccessibleObject GetSelected();
public System.Windows.Forms.AccessibleObject HitTest(
   int x,
   int y
);
```

DockEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class DockEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public DockEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

DocumentDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.ComponentModel.ToolboxItemFilterAttribute(FilterString = "System.Window
s.Forms", FilterType = Allow) */
public class DocumentDesigner
   extends System.Windows.Forms.Design.ScrollableControlDesigner
   implements System.ComponentModel.Design.IRootDesigner, System.Drawing.Design.IToolboxUser
```

Public Constructors

```
public DocumentDesigner();
```

Public Properties

```
public System.Windows.Forms.Design.SelectionRules get_SelectionRules();
```

Public Methods

```
public void Initialize(
   System.ComponentModel.IComponent component
);
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void EnsureMenuEditorService(
   System.ComponentModel.IComponent c
);
protected boolean GetToolSupported(
   System.Drawing.Design.ToolboxItem tool
);
protected void OnContextMenu(
   int x,
   int y
);
protected void OnCreateHandle();
protected void PreFilterProperties(
   System.Collections.IDictionary properties
);
protected void ToolPicked(
   System.Drawing.Design.ToolboxItem tool
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

Protected Fields

```
protected System.Windows.Forms.Design.IMenuEditorService menuEditorService;
```

EventHandlerService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EventHandlerService
   extends System.Object
   implements System.Windows.Forms.Design.IEventHandlerService
```

Public Constructors

```
public EventHandlerService(
   System.Windows.Forms.Control focusWnd
);
```

Public Properties

```
public final System.Windows.Forms.Control get_FocusWindow();
```

Public Methods

```
public final System.Object GetHandler(
   System.Type handlerType
);
public final void PopHandler(
   System.Object handler
);
public final void PushHandler(
   System.Object handler
);
```

Public Events

```
public final void add_EventHandlerChanged(
   System.EventHandler value
);
public final void remove_EventHandlerChanged(
   System.EventHandler value
);
```

EventsTab Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class EventsTab
   extends System.Windows.Forms.Design.PropertyTab
```

Public Constructors

```
public EventsTab(
   System.IServiceProvider sp
);
```

Public Properties

```
public System.String get_HelpKeyword();
public System.String get_TabName();
```

Public Methods

```
public boolean CanExtend(
   System.Object extendee
);
public System.ComponentModel.PropertyDescriptor GetDefaultProperty(
   System.Object obj
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.Object component,
   System.Attribute[] attributes
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object component,
   System.Attribute[] attributes
);
```

FileNameEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FileNameEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public FileNameEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

Protected Methods

```
protected void InitializeDialog(
   System.Windows.Forms.OpenFileDialog openFileDialog
);
```

FolderNameEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class FolderNameEditor
   extends System.Drawing.Design.UITypeEditor
```

Public Constructors

```
public FolderNameEditor();
```

Public Methods

```
public System.Object EditValue(
   System.ComponentModel.ITypeDescriptorContext context,
   System.IServiceProvider provider,
   System.Object value
);
public System.Drawing.Design.UITypeEditorEditStyle GetEditStyle(
   System.ComponentModel.ITypeDescriptorContext context
);
```

Protected Methods

```
protected void InitializeDialog(
   System.Windows.Forms.Design.FolderNameEditor+FolderBrowser folderBrowser
);
```

FolderNameEditor.FolderBrowser Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
protected class FolderBrowser
   extends System.ComponentModel.Component
```

Public Constructors

```
public FolderBrowser();
```

Public Properties

```
public System.String get_Description();
public void set_Description(
   System.String value
);
public System.String get_DirectoryPath();
public System.Windows.Forms.Design.FolderNameEditor+FolderBrowserFolder get_StartLocation();
public void set_StartLocation(
   System.Windows.Forms.Design.FolderNameEditor+FolderBrowserFolder value
);
public System.Windows.Forms.Design.FolderNameEditor+FolderBrowserStyles get_Style();
public void set_Style(
   System.Windows.Forms.Design.FolderNameEditor+FolderBrowserStyles value
);
```

Public Methods

```
public System.Windows.Forms.DialogResult ShowDialog();
public System.Windows.Forms.DialogResult ShowDialog(
   System.Windows.Forms.IWin32Window owner
);
```

FolderNameEditor.FolderBrowserFolder Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
protected enum FolderBrowserFolder
```

Members

Desktop

Favorites

MyComputer

MyDocuments

MyPictures

NetAndDialUpConnections

NetworkNeighborhood

Printers

Recent

SendTo

StartMenu

Templates

FolderNameEditor.FolderBrowserStyles Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
protected enum FolderBrowserStyles
```

Members

BrowseForComputer

BrowseForEverything

BrowseForPrinter

RestrictToDomain

RestrictToFilesystem

RestrictToSubfolders

ShowTextBox

IMenuEditorService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IMenuEditorService
```

Public Methods

```
System.Windows.Forms.Menu GetMenu();
boolean IsActive();
boolean MessageFilter(
   System.Windows.Forms.Message& m
);
void SetMenu(
   System.Windows.Forms.Menu menu
);
void SetSelection(
   System.Windows.Forms.MenuItem item
);
```

IUIService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Runtime.InteropServices.GuidAttribute(Value = "06A9C74B-5E32-4561-BE73-
381B37869F4F") */
public interface IUIService
```

Public Properties

```
System.Collections.IDictionary get_Styles();
```

Public Methods

```
boolean CanShowComponentEditor(
   System.Object component
);
System.Windows.Forms.IWin32Window GetDialogOwnerWindow();
void SetUIDirty();
boolean ShowComponentEditor(
   System.Object component,
   System.Windows.Forms.IWin32Window parent
);
System.Windows.Forms.DialogResult ShowDialog(
   System.Windows.Forms.Form form
);
void ShowError(
   System.Exception ex
);
void ShowError(
   System.String message
);
void ShowError(
   System.Exception ex,
   System.String message
);
void ShowMessage(
   System.String message
);
void ShowMessage(
   System.String message,
   System.String caption
);
System.Windows.Forms.DialogResult ShowMessage(
   System.String message,
   System.String caption,
   System.Windows.Forms.MessageBoxButtons buttons
);
boolean ShowToolWindow(
   System.Guid toolWindow
);
```

IWindowsFormsEditorService Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IWindowsFormsEditorService
```

Public Methods

```
void CloseDropDown();
void DropDownControl(
   System.Windows.Forms.Control control
);
System.Windows.Forms.DialogResult ShowDialog(
   System.Windows.Forms.Form dialog
);
```

MenuCommands Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class MenuCommands
   extends System.ComponentModel.Design.StandardCommands
```

Public Constructors

```
public MenuCommands();
```

Public Fields

```
public static final System.ComponentModel.Design.CommandID ComponentTrayMenu;
public static final System.ComponentModel.Design.CommandID ContainerMenu;
public static final System.ComponentModel.Design.CommandID DesignerProperties;
public static final System.ComponentModel.Design.CommandID KeyCancel;
public static final System.ComponentModel.Design.CommandID KeyDefaultAction;
public static final System.ComponentModel.Design.CommandID KeyMoveDown;
public static final System.ComponentModel.Design.CommandID KeyMoveLeft;
public static final System.ComponentModel.Design.CommandID KeyMoveRight;
public static final System.ComponentModel.Design.CommandID KeyMoveUp;
public static final System.ComponentModel.Design.CommandID KeyNudgeDown;
public static final System.ComponentModel.Design.CommandID KeyNudgeHeightDecrease;
public static final System.ComponentModel.Design.CommandID KeyNudgeHeightIncrease;
public static final System.ComponentModel.Design.CommandID KeyNudgeLeft;
public static final System.ComponentModel.Design.CommandID KeyNudgeRight;
public static final System.ComponentModel.Design.CommandID KeyNudgeUp;
public static final System.ComponentModel.Design.CommandID KeyNudgeWidthDecrease;
public static final System.ComponentModel.Design.CommandID KeyNudgeWidthIncrease;
public static final System.ComponentModel.Design.CommandID KeyReverseCancel;
public static final System.ComponentModel.Design.CommandID KeySelectNext;
public static final System.ComponentModel.Design.CommandID KeySelectPrevious;
public static final System.ComponentModel.Design.CommandID KeySizeHeightDecrease;
public static final System.ComponentModel.Design.CommandID KeySizeHeightIncrease;
public static final System.ComponentModel.Design.CommandID KeySizeWidthDecrease;
public static final System.ComponentModel.Design.CommandID KeySizeWidthIncrease;
public static final System.ComponentModel.Design.CommandID KeyTabOrderSelect;
public static final System.ComponentModel.Design.CommandID SelectionMenu;
public static final System.ComponentModel.Design.CommandID TraySelectionMenu;
```

ParentControlDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ParentControlDesigner
   extends System.Windows.Forms.Design.ControlDesigner
```

Public Constructors

```
public ParentControlDesigner();
```

Public Methods

```
public boolean CanParent(
   System.Windows.Forms.Control control
);
public boolean CanParent(
   System.Windows.Forms.Design.ControlDesigner controlDesigner
);
public void Initialize(
   System.ComponentModel.IComponent component
);
```

Protected Properties

```
protected System.Drawing.Point get_DefaultControlLocation();
protected boolean get_DrawGrid();
protected void set_DrawGrid(
   boolean value
);
protected boolean get_EnableDragRect();
protected System.Drawing.Size get_GridSize();
protected void set_GridSize(
   System.Drawing.Size value
);
```

Protected Methods

```
protected void CreateTool(
   System.Drawing.Design.ToolboxItem tool
);
protected void CreateTool(
   System.Drawing.Design.ToolboxItem tool,
   System.Drawing.Point location
);
protected void CreateTool(
   System.Drawing.Design.ToolboxItem tool,
   System.Drawing.Rectangle bounds
);
protected System.ComponentModel.IComponent[] CreateToolCore(
   System.Drawing.Design.ToolboxItem tool,
   int x,
   int y,
   int width,
   int height,
```

```
      boolean hasLocation,
      boolean hasSize
   );
   protected void Dispose(
      boolean disposing
   );
   protected System.Windows.Forms.Control GetControl(
      System.Object component
   );
   protected System.Drawing.Rectangle GetUpdatedRect(
      System.Drawing.Rectangle originalRect,
      System.Drawing.Rectangle dragRect,
      boolean updateSize
   );
   protected static void InvokeCreateTool(
      System.Windows.Forms.Design.ParentControlDesigner toInvoke,
      System.Drawing.Design.ToolboxItem tool
   );
   protected void OnDragDrop(
      System.Windows.Forms.DragEventArgs de
   );
   protected void OnDragEnter(
      System.Windows.Forms.DragEventArgs de
   );
   protected void OnDragLeave(
      System.EventArgs e
   );
   protected void OnDragOver(
      System.Windows.Forms.DragEventArgs de
   );
   protected void OnGiveFeedback(
      System.Windows.Forms.GiveFeedbackEventArgs e
   );
   protected void OnMouseDragBegin(
      int x,
      int y
   );
   protected void OnMouseDragEnd(
      boolean cancel
   );
   protected void OnMouseDragMove(
      int x,
      int y
   );
   protected void OnMouseEnter();
   protected void OnMouseHover();
   protected void OnMouseLeave();
   protected void OnPaintAdornments(
      System.Windows.Forms.PaintEventArgs pe
   );
   protected void OnSetCursor();
   protected void PreFilterProperties(
      System.Collections.IDictionary properties
   );
   protected void WndProc(
      System.Windows.Forms.Message& m
   );
```

PropertyTab Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class PropertyTab
   extends System.Object
   implements System.ComponentModel.IExtenderProvider
```

Public Properties

```
public System.Drawing.Bitmap get_Bitmap();
public System.Object[] get_Components();
public void set_Components(
   System.Object[] value
);
public System.String get_HelpKeyword();
public abstract System.String get_TabName();
```

Public Methods

```
public boolean CanExtend(
   System.Object extendee
);
public void Dispose();
public System.ComponentModel.PropertyDescriptor GetDefaultProperty(
   System.Object component
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.Object component
);
public abstract System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.Object component,
   System.Attribute[] attributes
);
public System.ComponentModel.PropertyDescriptorCollection GetProperties(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object component,
   System.Attribute[] attributes
);
```

Protected Constructors

```
protected PropertyTab();
```

Protected Methods

```
protected void Dispose(
   boolean disposing
);
protected void Finalize();
```

ScrollableControlDesigner Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ScrollableControlDesigner
   extends System.Windows.Forms.Design.ParentControlDesigner
```

Public Constructors

```
public ScrollableControlDesigner();
```

Protected Methods

```
protected boolean GetHitTest(
   System.Drawing.Point pt
);
protected void WndProc(
   System.Windows.Forms.Message& m
);
```

SelectionRules Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum SelectionRules
```

Members

AllSizeable

BottomSizeable

LeftSizeable

Locked

Moveable

None

RightSizeable

TopSizeable

Visible

WindowsFormsComponentEditor Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class WindowsFormsComponentEditor
   extends System.ComponentModel.ComponentEditor
```

Public Methods

```
public boolean EditComponent(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object component
);
public boolean EditComponent(
   System.Object component,
   System.Windows.Forms.IWin32Window owner
);
public boolean EditComponent(
   System.ComponentModel.ITypeDescriptorContext context,
   System.Object component,
   System.Windows.Forms.IWin32Window owner
);
```

Protected Constructors

```
protected WindowsFormsComponentEditor();
```

Protected Methods

```
protected System.Type[] GetComponentEditorPages();
protected int GetInitialComponentEditorPageIndex();
```

System.Xml Namespace (Visual J# Syntax)

Classes

NameTable

XmlAttribute

XmlAttributeCollection

XmlCDataSection

XmlCharacterData

XmlComment

XmlConvert

XmlDataDocument

XmlDeclaration

XmlDocument

XmlDocumentFragment

XmlDocumentType

XmlElement

XmlEntity

XmlEntityReference

XmlException

XmlImplementation

XmlLinkedNode

XmlNamedNodeMap

XmlNamespaceManager

XmlNameTable

XmlNode

XmlNodeChangedEventArgs

XmlNodeList

XmlNodeReader

XmlNotation

XmlParserContext

XmlProcessingInstruction

XmlQualifiedName

XmlReader

XmlResolver

XmlSignificantWhitespace

XmlText

XmlTextReader

XmlTextWriter

XmlUrlResolver

XmlValidatingReader

XmlWhitespace

XmlWriter

Interfaces

IHasXmlNode

IXmlLineInfo

Delegates

XmlNodeChangedEventHandler

Enumerations

EntityHandling

Formatting

ReadState

ValidationType

WhitespaceHandling

WriteState

XmlNodeChangedAction

XmlNodeOrder

XmlNodeType

XmlSpace

XmlTokenizedType

EntityHandling Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum EntityHandling
```

Members

ExpandCharEntities

ExpandEntities

Formatting Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum Formatting
```

Members

Indented

None

IHasXmlNode Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IHasXmlNode
```

Public Methods

```
System.Xml.XmlNode GetNode();
```

IXmlLineInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IXmlLineInfo
```

Public Properties

```
int get_LineNumber();
int get_LinePosition();
```

Public Methods

```
boolean HasLineInfo();
```

NameTable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class NameTable
   extends System.Xml.XmlNameTable
```

Public Constructors

```
public NameTable();
```

Public Methods

```
public System.String Add(
   System.String key
);
public System.String Add(
   char[] key,
   int start,
   int len
);
public System.String Get(
   System.String value
);
public System.String Get(
   char[] key,
   int start,
   int len
);
```

ReadState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ReadState
```

Members

Closed

EndOfFile

Error

Initial

Interactive

ValidationType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum ValidationType
```

Members

Auto

DTD

None

Schema

XDR

WhitespaceHandling Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum WhitespaceHandling
```

Members

All

None

Significant

WriteState Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum WriteState
```

Members

Attribute

Closed

Content

Element

Prolog

Start

XmlAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlAttribute
   extends System.Xml.XmlNode
```

Public Properties

```
public System.String get_BaseURI();
public System.String get_InnerText();
public void set_InnerText(
   System.String value
);
public System.String get_InnerXml();
public void set_InnerXml(
   System.String value
);
public System.String get_LocalName();
public System.String get_Name();
public System.String get_NamespaceURI();
public System.Xml.XmlNodeType get_NodeType();
public System.Xml.XmlDocument get_OwnerDocument();
public System.Xml.XmlElement get_OwnerElement();
public System.Xml.XmlNode get_ParentNode();
public System.String get_Prefix();
public void set_Prefix(
   System.String value
);
public boolean get_Specified();
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public void WriteContentTo(
   System.Xml.XmlWriter w
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

Protected Constructors

```
protected XmlAttribute(
   System.String prefix,
   System.String localName,
   System.String namespaceURI,
   System.Xml.XmlDocument doc
);
```

XmlAttributeCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "ItemOf") */
public class XmlAttributeCollection
   extends System.Xml.XmlNamedNodeMap
   implements System.Collections.ICollection
```

Public Properties

```
public System.Xml.XmlAttribute get_ItemOf(
   int i
);
public System.Xml.XmlAttribute get_ItemOf(
   System.String name
);
public System.Xml.XmlAttribute get_ItemOf(
   System.String localName,
   System.String namespaceURI
);
```

Public Methods

```
public System.Xml.XmlAttribute Append(
   System.Xml.XmlAttribute node
);
public void CopyTo(
   System.Xml.XmlAttribute[] array,
   int index
);
public System.Xml.XmlAttribute InsertAfter(
   System.Xml.XmlAttribute newNode,
   System.Xml.XmlAttribute refNode
);
public System.Xml.XmlAttribute InsertBefore(
   System.Xml.XmlAttribute newNode,
   System.Xml.XmlAttribute refNode
);
public System.Xml.XmlAttribute Prepend(
   System.Xml.XmlAttribute node
);
public System.Xml.XmlAttribute Remove(
   System.Xml.XmlAttribute node
);
public void RemoveAll();
public System.Xml.XmlAttribute RemoveAt(
   int i
);
public System.Xml.XmlNode SetNamedItem(
   System.Xml.XmlNode node
);
```

XmlCDataSection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlCDataSection
   extends System.Xml.XmlCharacterData
```

Public Properties

```
public System.String get_LocalName();
public System.String get_Name();
public System.Xml.XmlNodeType get_NodeType();
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public void WriteContentTo(
   System.Xml.XmlWriter w
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

Protected Constructors

```
protected XmlCDataSection(
   System.String data,
   System.Xml.XmlDocument doc
);
```

XmlCharacterData Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlCharacterData
   extends System.Xml.XmlLinkedNode
```

Public Properties

```
public System.String get_Data();
public void set_Data(
   System.String value
);
public System.String get_InnerText();
public void set_InnerText(
   System.String value
);
public int get_Length();
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Methods

```
public void AppendData(
   System.String strData
);
public void DeleteData(
   int offset,
   int count
);
public void InsertData(
   int offset,
   System.String strData
);
public void ReplaceData(
   int offset,
   int count,
   System.String strData
);
public System.String Substring(
   int offset,
   int count
);
```

Protected Constructors

```
protected XmlCharacterData(
   System.String data,
   System.Xml.XmlDocument doc
);
```

XmlComment Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlComment
   extends System.Xml.XmlCharacterData
```

Public Properties

```
public System.String get_LocalName();
public System.String get_Name();
public System.Xml.XmlNodeType get_NodeType();
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public void WriteContentTo(
   System.Xml.XmlWriter w
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

Protected Constructors

```
protected XmlComment(
   System.String comment,
   System.Xml.XmlDocument doc
);
```

XmlConvert Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlConvert
   extends System.Object
```

Public Constructors

```
public XmlConvert();
```

Public Methods

```
public static System.String DecodeName(
   System.String name
);
public static System.String EncodeLocalName(
   System.String name
);
public static System.String EncodeName(
   System.String name
);
public static System.String EncodeNmToken(
   System.String name
);
public static boolean ToBoolean(
   System.String s
);
public static ubyte ToByte(
   System.String s
);
public static char ToChar(
   System.String s
);
public static System.DateTime ToDateTime(
   System.String s
);
public static System.DateTime ToDateTime(
   System.String s,
   System.String format
);
public static System.DateTime ToDateTime(
   System.String s,
   System.String[] formats
);
public static System.Decimal ToDecimal(
   System.String s
);
public static double ToDouble(
   System.String s
);
public static System.Guid ToGuid(
   System.String s
);
public static short ToInt16(
   System.String s
);
public static int ToInt32(
   System.String s
```

```
);
public static long ToInt64(
   System.String s
);
public static float ToSingle(
   System.String s
);
public static System.String ToString(
   boolean value
);
public static System.String ToString(
   ubyte value
);
public static System.String ToString(
   char value
);
public static System.String ToString(
   double value
);
public static System.String ToString(
   short value
);
public static System.String ToString(
   int value
);
public static System.String ToString(
   long value
);
public static System.String ToString(
   float value
);
public static System.String ToString(
   System.DateTime value
);
public static System.String ToString(
   System.Decimal value
);
public static System.String ToString(
   System.Guid value
);
public static System.String ToString(
   System.TimeSpan value
);
public static System.String ToString(
   System.DateTime value,
   System.String format
);
public static System.TimeSpan ToTimeSpan(
   System.String s
);
public static System.String VerifyName(
   System.String name
);
public static System.String VerifyNCName(
   System.String name
);
```

XmlDataDocument Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlDataDocument
   extends System.Xml.XmlDocument
```

Public Constructors

```
public XmlDataDocument();
public XmlDataDocument(
   System.Data.DataSet dataset
);
```

Public Properties

```
public System.Data.DataSet get_DataSet();
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public System.Xml.XmlElement CreateElement(
   System.String prefix,
   System.String localName,
   System.String namespaceURI
);
public System.Xml.XmlEntityReference CreateEntityReference(
   System.String name
);
public System.Xml.XmlElement GetElementById(
   System.String elemId
);
public System.Xml.XmlElement GetElementFromRow(
   System.Data.DataRow r
);
public System.Data.DataRow GetRowFromElement(
   System.Xml.XmlElement e
);
public void Load(
   System.IO.Stream inStream
);
public void Load(
   System.IO.TextReader txtReader
);
public void Load(
   System.String filename
);
public void Load(
   System.Xml.XmlReader reader
);
```

Protected Methods

```
protected System.Xml.XPath.XPathNavigator CreateNavigator(
   System.Xml.XmlNode node
);
```

XmlDeclaration Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlDeclaration
   extends System.Xml.XmlLinkedNode
```

Public Properties

```
public System.String get_Encoding();
public void set_Encoding(
   System.String value
);
public System.String get_InnerText();
public void set_InnerText(
   System.String value
);
public System.String get_LocalName();
public System.String get_Name();
public System.Xml.XmlNodeType get_NodeType();
public System.String get_Standalone();
public void set_Standalone(
   System.String value
);
public System.String get_Value();
public void set_Value(
   System.String value
);
public System.String get_Version();
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public void WriteContentTo(
   System.Xml.XmlWriter w
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

Protected Constructors

```
protected XmlDeclaration(
   System.String version,
   System.String encoding,
   System.String standalone,
   System.Xml.XmlDocument doc
);
```

XmlDocument Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlDocument
   extends System.Xml.XmlNode
```

Public Constructors

```
public XmlDocument();
public XmlDocument(
   System.Xml.XmlNameTable nt
);
```

Public Properties

```
public System.String get_BaseURI();
public System.Xml.XmlElement get_DocumentElement();
public System.Xml.XmlDocumentType get_DocumentType();
public System.Xml.XmlImplementation get_Implementation();
public System.String get_InnerXml();
public void set_InnerXml(
   System.String value
);
public boolean get_IsReadOnly();
public System.String get_LocalName();
public System.String get_Name();
public System.Xml.XmlNameTable get_NameTable();
public System.Xml.XmlNodeType get_NodeType();
public System.Xml.XmlDocument get_OwnerDocument();
public boolean get_PreserveWhitespace();
public void set_PreserveWhitespace(
   boolean value
);
public void set_XmlResolver(
   System.Xml.XmlResolver value
);
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public System.Xml.XmlAttribute CreateAttribute(
   System.String name
);
public System.Xml.XmlAttribute CreateAttribute(
   System.String qualifiedName,
   System.String namespaceURI
);
public System.Xml.XmlAttribute CreateAttribute(
   System.String prefix,
   System.String localName,
   System.String namespaceURI
);
public System.Xml.XmlCDataSection CreateCDataSection(
   System.String data
```

```
);
public System.Xml.XmlComment CreateComment(
   System.String data
);
public System.Xml.XmlDocumentFragment CreateDocumentFragment();
public System.Xml.XmlDocumentType CreateDocumentType(
   System.String name,
   System.String publicId,
   System.String systemId,
   System.String internalSubset
);
public System.Xml.XmlElement CreateElement(
   System.String name
);
public System.Xml.XmlElement CreateElement(
   System.String qualifiedName,
   System.String namespaceURI
);
public System.Xml.XmlElement CreateElement(
   System.String prefix,
   System.String localName,
   System.String namespaceURI
);
public System.Xml.XmlEntityReference CreateEntityReference(
   System.String name
);
public System.Xml.XmlNode CreateNode(
   System.String nodeTypeString,
   System.String name,
   System.String namespaceURI
);
public System.Xml.XmlNode CreateNode(
   System.Xml.XmlNodeType type,
   System.String name,
   System.String namespaceURI
);
public System.Xml.XmlNode CreateNode(
   System.Xml.XmlNodeType type,
   System.String prefix,
   System.String name,
   System.String namespaceURI
);
public System.Xml.XmlProcessingInstruction CreateProcessingInstruction(
   System.String target,
   System.String data
);
public System.Xml.XmlSignificantWhitespace CreateSignificantWhitespace(
   System.String text
);
public System.Xml.XmlText CreateTextNode(
   System.String text
);
public System.Xml.XmlWhitespace CreateWhitespace(
   System.String text
);
public System.Xml.XmlDeclaration CreateXmlDeclaration(
   System.String version,
   System.String encoding,
   System.String standalone
);
public System.Xml.XmlElement GetElementById(
   System.String elementId
);
public System.Xml.XmlNodeList GetElementsByTagName(
   System.String name
);
public System.Xml.XmlNodeList GetElementsByTagName(
   System.String localName,
```

```
   System.String namespaceURI
);
public System.Xml.XmlNode ImportNode(
   System.Xml.XmlNode node,
   boolean deep
);
public void Load(
   System.IO.Stream inStream
);
public void Load(
   System.IO.TextReader txtReader
);
public void Load(
   System.String filename
);
public void Load(
   System.Xml.XmlReader reader
);
public void LoadXml(
   System.String xml
);
public System.Xml.XmlNode ReadNode(
   System.Xml.XmlReader reader
);
public void Save(
   System.IO.Stream outStream
);
public void Save(
   System.IO.TextWriter writer
);
public void Save(
   System.String filename
);
public void Save(
   System.Xml.XmlWriter w
);
public void WriteContentTo(
   System.Xml.XmlWriter xw
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

Public Events

```
public void add_NodeChanged(
   System.Xml.XmlNodeChangedEventHandler value
);
public void remove_NodeChanged(
   System.Xml.XmlNodeChangedEventHandler value
);
public void add_NodeChanging(
   System.Xml.XmlNodeChangedEventHandler value
);
public void remove_NodeChanging(
   System.Xml.XmlNodeChangedEventHandler value
);
public void add_NodeInserted(
   System.Xml.XmlNodeChangedEventHandler value
);
public void remove_NodeInserted(
   System.Xml.XmlNodeChangedEventHandler value
);
public void add_NodeInserting(
   System.Xml.XmlNodeChangedEventHandler value
);
```

```
public void remove_NodeInserting(
   System.Xml.XmlNodeChangedEventHandler value
);
public void add_NodeRemoved(
   System.Xml.XmlNodeChangedEventHandler value
);
public void remove_NodeRemoved(
   System.Xml.XmlNodeChangedEventHandler value
);
public void add_NodeRemoving(
   System.Xml.XmlNodeChangedEventHandler value
);
public void remove_NodeRemoving(
   System.Xml.XmlNodeChangedEventHandler value
);
```

Protected Constructors

```
protected XmlDocument(
   System.Xml.XmlImplementation imp
);
```

Protected Methods

```
protected System.Xml.XmlAttribute CreateDefaultAttribute(
   System.String prefix,
   System.String localName,
   System.String namespaceURI
);
protected System.Xml.XPath.XPathNavigator CreateNavigator(
   System.Xml.XmlNode node
);
```

XmlDocumentFragment Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlDocumentFragment
   extends System.Xml.XmlNode
```

Public Properties

```
public System.String get_InnerXml();
public void set_InnerXml(
   System.String value
);
public System.String get_LocalName();
public System.String get_Name();
public System.Xml.XmlNodeType get_NodeType();
public System.Xml.XmlDocument get_OwnerDocument();
public System.Xml.XmlNode get_ParentNode();
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public void WriteContentTo(
   System.Xml.XmlWriter w
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

Protected Constructors

```
protected XmlDocumentFragment(
   System.Xml.XmlDocument ownerDocument
);
```

XmlDocumentType Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlDocumentType
   extends System.Xml.XmlLinkedNode
```

Public Properties

```
public System.Xml.XmlNamedNodeMap get_Entities();
public System.String get_InternalSubset();
public boolean get_IsReadOnly();
public System.String get_LocalName();
public System.String get_Name();
public System.Xml.XmlNodeType get_NodeType();
public System.Xml.XmlNamedNodeMap get_Notations();
public System.String get_PublicId();
public System.String get_SystemId();
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public void WriteContentTo(
   System.Xml.XmlWriter w
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

Protected Constructors

```
protected XmlDocumentType(
   System.String name,
   System.String publicId,
   System.String systemId,
   System.String internalSubset,
   System.Xml.XmlDocument doc
);
```

XmlElement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlElement
   extends System.Xml.XmlLinkedNode
```

Public Properties

```
public System.Xml.XmlAttributeCollection get_Attributes();
public boolean get_HasAttributes();
public System.String get_InnerText();
public void set_InnerText(
   System.String value
);
public System.String get_InnerXml();
public void set_InnerXml(
   System.String value
);
public boolean get_IsEmpty();
public void set_IsEmpty(
   boolean value
);
public System.String get_LocalName();
public System.String get_Name();
public System.String get_NamespaceURI();
public System.Xml.XmlNode get_NextSibling();
public System.Xml.XmlNodeType get_NodeType();
public System.Xml.XmlDocument get_OwnerDocument();
public System.String get_Prefix();
public void set_Prefix(
   System.String value
);
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public System.String GetAttribute(
   System.String name
);
public System.String GetAttribute(
   System.String localName,
   System.String namespaceURI
);
public System.Xml.XmlAttribute GetAttributeNode(
   System.String name
);
public System.Xml.XmlAttribute GetAttributeNode(
   System.String localName,
   System.String namespaceURI
);
public System.Xml.XmlNodeList GetElementsByTagName(
   System.String name
);
public System.Xml.XmlNodeList GetElementsByTagName(
   System.String localName,
   System.String namespaceURI
```

```
);
public boolean HasAttribute(
   System.String name
);
public boolean HasAttribute(
   System.String localName,
   System.String namespaceURI
);
public void RemoveAll();
public void RemoveAllAttributes();
public void RemoveAttribute(
   System.String name
);
public void RemoveAttribute(
   System.String localName,
   System.String namespaceURI
);
public System.Xml.XmlNode RemoveAttributeAt(
   int i
);
public System.Xml.XmlAttribute RemoveAttributeNode(
   System.Xml.XmlAttribute oldAttr
);
public System.Xml.XmlAttribute RemoveAttributeNode(
   System.String localName,
   System.String namespaceURI
);
public void SetAttribute(
   System.String name,
   System.String value
);
public System.String SetAttribute(
   System.String localName,
   System.String namespaceURI,
   System.String value
);
public System.Xml.XmlAttribute SetAttributeNode(
   System.Xml.XmlAttribute newAttr
);
public System.Xml.XmlAttribute SetAttributeNode(
   System.String localName,
   System.String namespaceURI
);
public void WriteContentTo(
   System.Xml.XmlWriter w
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

Protected Constructors

```
protected XmlElement(
   System.String prefix,
   System.String localName,
   System.String namespaceURI,
   System.Xml.XmlDocument doc
);
```

XmlEntity Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlEntity
   extends System.Xml.XmlNode
```

Public Properties

```
public System.String get_BaseURI();
public System.String get_InnerText();
public void set_InnerText(
   System.String value
);
public System.String get_InnerXml();
public void set_InnerXml(
   System.String value
);
public boolean get_IsReadOnly();
public System.String get_LocalName();
public System.String get_Name();
public System.Xml.XmlNodeType get_NodeType();
public System.String get_NotationName();
public System.String get_OuterXml();
public System.String get_PublicId();
public System.String get_SystemId();
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public void WriteContentTo(
   System.Xml.XmlWriter w
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

XmlEntityReference Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlEntityReference
   extends System.Xml.XmlLinkedNode
```

Public Properties

```
public System.String get_BaseURI();
public boolean get_IsReadOnly();
public System.String get_LocalName();
public System.String get_Name();
public System.Xml.XmlNodeType get_NodeType();
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public void WriteContentTo(
   System.Xml.XmlWriter w
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

Protected Constructors

```
protected XmlEntityReference(
   System.String name,
   System.Xml.XmlDocument doc
);
```

XmlException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class XmlException
   extends System.SystemException
```

Public Constructors

```
public XmlException(
   System.String message,
   System.Exception innerException
);
```

Public Properties

```
public int get_LineNumber();
public int get_LinePosition();
public System.String get_Message();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected XmlException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

XmlImplementation Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlImplementation
   extends System.Object
```

Public Constructors

```
public XmlImplementation();
```

Public Methods

```
public System.Xml.XmlDocument CreateDocument();
public boolean HasFeature(
   System.String strFeature,
   System.String strVersion
);
```

XmlLinkedNode Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlLinkedNode
   extends System.Xml.XmlNode
```

Public Properties

```
public System.Xml.XmlNode get_NextSibling();
public System.Xml.XmlNode get_PreviousSibling();
```

XmlNamedNodeMap Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlNamedNodeMap
   extends System.Object
   implements System.Collections.IEnumerable
```

Public Properties

```
public int get_Count();
```

Public Methods

```
public System.Collections.IEnumerator GetEnumerator();
public System.Xml.XmlNode GetNamedItem(
   System.String name
);
public System.Xml.XmlNode GetNamedItem(
   System.String localName,
   System.String namespaceURI
);
public System.Xml.XmlNode Item(
   int index
);
public System.Xml.XmlNode RemoveNamedItem(
   System.String name
);
public System.Xml.XmlNode RemoveNamedItem(
   System.String localName,
   System.String namespaceURI
);
public System.Xml.XmlNode SetNamedItem(
   System.Xml.XmlNode node
);
```

XmlNamespaceManager Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlNamespaceManager
   extends System.Object
   implements System.Collections.IEnumerable
```

Public Constructors

```
public XmlNamespaceManager(
   System.Xml.XmlNameTable nameTable
);
```

Public Properties

```
public System.String get_DefaultNamespace();
public System.Xml.XmlNameTable get_NameTable();
```

Public Methods

```
public void AddNamespace(
   System.String prefix,
   System.String uri
);
public System.Collections.IEnumerator GetEnumerator();
public boolean HasNamespace(
   System.String prefix
);
public System.String LookupNamespace(
   System.String prefix
);
public System.String LookupPrefix(
   System.String uri
);
public boolean PopScope();
public void PushScope();
public void RemoveNamespace(
   System.String prefix,
   System.String uri
);
```

XmlNameTable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlNameTable
   extends System.Object
```

Public Methods

```
public abstract System.String Add(
   System.String array
);
public abstract System.String Add(
   char[] array,
   int offset,
   int length
);
public abstract System.String Get(
   System.String array
);
public abstract System.String Get(
   char[] array,
   int offset,
   int length
);
```

Protected Constructors

```
protected XmlNameTable();
```

XmlNode Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public abstract class XmlNode
   extends System.Object
   implements System.ICloneable, System.Collections.IEnumerable, System.Xml.XPath.IXPathNavig
able
```

Public Properties

```
public System.Xml.XmlAttributeCollection get_Attributes();
public System.String get_BaseURI();
public System.Xml.XmlNodeList get_ChildNodes();
public System.Xml.XmlNode get_FirstChild();
public boolean get_HasChildNodes();
public System.String get_InnerText();
public void set_InnerText(
   System.String value
);
public System.String get_InnerXml();
public void set_InnerXml(
   System.String value
);
public boolean get_IsReadOnly();
public System.Xml.XmlElement get_Item(
   System.String name
);
public System.Xml.XmlElement get_Item(
   System.String localname,
   System.String ns
);
public System.Xml.XmlNode get_LastChild();
public abstract System.String get_LocalName();
public abstract System.String get_Name();
public System.String get_NamespaceURI();
public System.Xml.XmlNode get_NextSibling();
public abstract System.Xml.XmlNodeType get_NodeType();
public System.String get_OuterXml();
public System.Xml.XmlDocument get_OwnerDocument();
public System.Xml.XmlNode get_ParentNode();
public System.String get_Prefix();
public void set_Prefix(
   System.String value
);
public System.Xml.XmlNode get_PreviousSibling();
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Methods

```
public System.Xml.XmlNode AppendChild(
   System.Xml.XmlNode newChild
);
public System.Xml.XmlNode Clone();
public abstract System.Xml.XmlNode CloneNode(
```

```
   boolean deep
);
public final System.Xml.XPath.XPathNavigator CreateNavigator();
public System.Collections.IEnumerator GetEnumerator();
public System.String GetNamespaceOfPrefix(
   System.String prefix
);
public System.String GetPrefixOfNamespace(
   System.String namespaceURI
);
public System.Xml.XmlNode InsertAfter(
   System.Xml.XmlNode newChild,
   System.Xml.XmlNode refChild
);
public System.Xml.XmlNode InsertBefore(
   System.Xml.XmlNode newChild,
   System.Xml.XmlNode refChild
);
public void Normalize();
public System.Xml.XmlNode PrependChild(
   System.Xml.XmlNode newChild
);
public void RemoveAll();
public System.Xml.XmlNode RemoveChild(
   System.Xml.XmlNode oldChild
);
public System.Xml.XmlNode ReplaceChild(
   System.Xml.XmlNode newChild,
   System.Xml.XmlNode oldChild
);
public System.Xml.XmlNodeList SelectNodes(
   System.String xpath
);
public System.Xml.XmlNodeList SelectNodes(
   System.String xpath,
   System.Xml.XmlNamespaceManager nsmgr
);
public System.Xml.XmlNode SelectSingleNode(
   System.String xpath
);
public System.Xml.XmlNode SelectSingleNode(
   System.String xpath,
   System.Xml.XmlNamespaceManager nsmgr
);
public boolean Supports(
   System.String feature,
   System.String version
);
public abstract void WriteContentTo(
   System.Xml.XmlWriter w
);
public abstract void WriteTo(
   System.Xml.XmlWriter w
);
```

XmlNodeChangedAction Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlNodeChangedAction
```

Members

Change

Insert

Remove

XmlNodeChangedEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlNodeChangedEventArgs
   extends System.Object
```

Public Properties

```
public System.Xml.XmlNodeChangedAction get_Action();
public System.Xml.XmlNode get_NewParent();
public System.Xml.XmlNode get_Node();
public System.Xml.XmlNode get_OldParent();
```

XmlNodeChangedEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void XmlNodeChangedEventHandler(
   System.Object sender,
   System.Xml.XmlNodeChangedEventArgs e
);
```

XmlNodeList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "ItemOf") */
public abstract class XmlNodeList
   extends System.Object
   implements System.Collections.IEnumerable
```

Public Properties

```
public abstract int get_Count();
public System.Xml.XmlNode get_ItemOf(
   int i
);
```

Public Methods

```
public abstract System.Collections.IEnumerator GetEnumerator();
public abstract System.Xml.XmlNode Item(
   int index
);
```

Protected Constructors

```
protected XmlNodeList();
```

XmlNodeOrder Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlNodeOrder
```

Members

After

Before

Same

Unknown

XmlNodeReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class XmlNodeReader
   extends System.Xml.XmlReader
```

Public Constructors

```
public XmlNodeReader(
   System.Xml.XmlNode node
);
```

Public Properties

```
public int get_AttributeCount();
public System.String get_BaseURI();
public boolean get_CanResolveEntity();
public int get_Depth();
public boolean get_EOF();
public boolean get_HasAttributes();
public boolean get_HasValue();
public boolean get_IsDefault();
public boolean get_IsEmptyElement();
public System.String get_Item(
   int i
);
public System.String get_Item(
   System.String name
);
public System.String get_Item(
   System.String name,
   System.String namespaceURI
);
public System.String get_LocalName();
public System.String get_Name();
public System.String get_NamespaceURI();
public System.Xml.XmlNameTable get_NameTable();
public System.Xml.XmlNodeType get_NodeType();
public System.String get_Prefix();
public char get_QuoteChar();
public System.Xml.ReadState get_ReadState();
public System.String get_Value();
public System.String get_XmlLang();
public System.Xml.XmlSpace get_XmlSpace();
```

Public Methods

```
public void Close();
public System.String GetAttribute(
   int attributeIndex
);
public System.String GetAttribute(
   System.String name
);
public System.String GetAttribute(
```

```
      System.String name,
      System.String namespaceURI
   );
   public System.String LookupNamespace(
      System.String prefix
   );
   public boolean MoveToAttribute(
      System.String name
   );
   public void MoveToAttribute(
      int attributeIndex
   );
   public boolean MoveToAttribute(
      System.String name,
      System.String namespaceURI
   );
   public boolean MoveToElement();
   public boolean MoveToFirstAttribute();
   public boolean MoveToNextAttribute();
   public boolean Read();
   public boolean ReadAttributeValue();
   public System.String ReadInnerXml();
   public System.String ReadOuterXml();
   public System.String ReadString();
   public void ResolveEntity();
   public void Skip();
```

XmlNodeType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlNodeType
```

Members

Attribute

CDATA

Comment

Document

DocumentFragment

DocumentType

Element

EndElement

EndEntity

Entity

EntityReference

None

Notation

ProcessingInstruction

SignificantWhitespace

Text

Whitespace

XmlDeclaration

XmlNotation Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlNotation
   extends System.Xml.XmlNode
```

Public Properties

```
public System.String get_InnerXml();
public void set_InnerXml(
   System.String value
);
public boolean get_IsReadOnly();
public System.String get_LocalName();
public System.String get_Name();
public System.Xml.XmlNodeType get_NodeType();
public System.String get_OuterXml();
public System.String get_PublicId();
public System.String get_SystemId();
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public void WriteContentTo(
   System.Xml.XmlWriter w
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

XmlParserContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlParserContext
   extends System.Object
```

Public Constructors

```
public XmlParserContext(
   System.Xml.XmlNameTable nt,
   System.Xml.XmlNamespaceManager nsMgr,
   System.String xmlLang,
   System.Xml.XmlSpace xmlSpace
);
public XmlParserContext(
   System.Xml.XmlNameTable nt,
   System.Xml.XmlNamespaceManager nsMgr,
   System.String xmlLang,
   System.Xml.XmlSpace xmlSpace,
   System.Text.Encoding enc
);
public XmlParserContext(
   System.Xml.XmlNameTable nt,
   System.Xml.XmlNamespaceManager nsMgr,
   System.String docTypeName,
   System.String pubId,
   System.String sysId,
   System.String internalSubset,
   System.String baseURI,
   System.String xmlLang,
   System.Xml.XmlSpace xmlSpace
);
public XmlParserContext(
   System.Xml.XmlNameTable nt,
   System.Xml.XmlNamespaceManager nsMgr,
   System.String docTypeName,
   System.String pubId,
   System.String sysId,
   System.String internalSubset,
   System.String baseURI,
   System.String xmlLang,
   System.Xml.XmlSpace xmlSpace,
   System.Text.Encoding enc
);
```

Public Properties

```
public System.String get_BaseURI();
public void set_BaseURI(
   System.String value
);
public System.String get_DocTypeName();
public void set_DocTypeName(
   System.String value
);
public System.Text.Encoding get_Encoding();
public void set_Encoding(
   System.Text.Encoding value
```

```
);
public System.String get_InternalSubset();
public void set_InternalSubset(
   System.String value
);
public System.Xml.XmlNamespaceManager get_NamespaceManager();
public void set_NamespaceManager(
   System.Xml.XmlNamespaceManager value
);
public System.Xml.XmlNameTable get_NameTable();
public void set_NameTable(
   System.Xml.XmlNameTable value
);
public System.String get_PublicId();
public void set_PublicId(
   System.String value
);
public System.String get_SystemId();
public void set_SystemId(
   System.String value
);
public System.String get_XmlLang();
public void set_XmlLang(
   System.String value
);
public System.Xml.XmlSpace get_XmlSpace();
public void set_XmlSpace(
   System.Xml.XmlSpace value
);
```

XmlProcessingInstruction Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlProcessingInstruction
   extends System.Xml.XmlLinkedNode
```

Public Properties

```
public System.String get_Data();
public void set_Data(
   System.String value
);
public System.String get_InnerText();
public void set_InnerText(
   System.String value
);
public System.String get_LocalName();
public System.String get_Name();
public System.Xml.XmlNodeType get_NodeType();
public System.String get_Target();
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public void WriteContentTo(
   System.Xml.XmlWriter w
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

Protected Constructors

```
protected XmlProcessingInstruction(
   System.String target,
   System.String data,
   System.Xml.XmlDocument doc
);
```

XmlQualifiedName Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlQualifiedName
   extends System.Object
```

Public Constructors

```
public XmlQualifiedName();
public XmlQualifiedName(
   System.String name
);
public XmlQualifiedName(
   System.String name,
   System.String ns
);
```

Public Properties

```
public boolean get_IsEmpty();
public System.String get_Name();
public System.String get_Namespace();
```

Public Methods

```
public boolean Equals(
   System.Object other
);
public int GetHashCode();
public System.String ToString();
public static System.String ToString(
   System.String name,
   System.String ns
);
```

Public Operators

```
public static boolean op_Equality(
   System.Xml.XmlQualifiedName a,
   System.Xml.XmlQualifiedName b
);
public static boolean op_Inequality(
   System.Xml.XmlQualifiedName a,
   System.Xml.XmlQualifiedName b
);
```

Public Fields

```
public static final System.Xml.XmlQualifiedName Empty;
```

XmlReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public abstract class XmlReader
   extends System.Object
```

Public Properties

```
public abstract int get_AttributeCount();
public abstract System.String get_BaseURI();
public boolean get_CanResolveEntity();
public abstract int get_Depth();
public abstract boolean get_EOF();
public boolean get_HasAttributes();
public abstract boolean get_HasValue();
public abstract boolean get_IsDefault();
public abstract boolean get_IsEmptyElement();
public abstract System.String get_Item(
   int i
);
public abstract System.String get_Item(
   System.String name
);
public abstract System.String get_Item(
   System.String name,
   System.String namespaceURI
);
public abstract System.String get_LocalName();
public abstract System.String get_Name();
public abstract System.String get_NamespaceURI();
public abstract System.Xml.XmlNameTable get_NameTable();
public abstract System.Xml.XmlNodeType get_NodeType();
public abstract System.String get_Prefix();
public abstract char get_QuoteChar();
public abstract System.Xml.ReadState get_ReadState();
public abstract System.String get_Value();
public abstract System.String get_XmlLang();
public abstract System.Xml.XmlSpace get_XmlSpace();
```

Public Methods

```
public abstract void Close();
public abstract System.String GetAttribute(
   int i
);
public abstract System.String GetAttribute(
   System.String name
);
public abstract System.String GetAttribute(
   System.String name,
   System.String namespaceURI
);
public static boolean IsName(
   System.String str
);
public static boolean IsNameToken(
   System.String str
```

```
);
public boolean IsStartElement();
public boolean IsStartElement(
   System.String name
);
public boolean IsStartElement(
   System.String localname,
   System.String ns
);
public abstract System.String LookupNamespace(
   System.String prefix
);
public abstract boolean MoveToAttribute(
   System.String name
);
public abstract void MoveToAttribute(
   int i
);
public abstract boolean MoveToAttribute(
   System.String name,
   System.String ns
);
public System.Xml.XmlNodeType MoveToContent();
public abstract boolean MoveToElement();
public abstract boolean MoveToFirstAttribute();
public abstract boolean MoveToNextAttribute();
public abstract boolean Read();
public abstract boolean ReadAttributeValue();
public System.String ReadElementString();
public System.String ReadElementString(
   System.String name
);
public System.String ReadElementString(
   System.String localname,
   System.String ns
);
public void ReadEndElement();
public abstract System.String ReadInnerXml();
public abstract System.String ReadOuterXml();
public void ReadStartElement();
public void ReadStartElement(
   System.String name
);
public void ReadStartElement(
   System.String localname,
   System.String ns
);
public abstract System.String ReadString();
public abstract void ResolveEntity();
public void Skip();
```

Protected Constructors

```
protected XmlReader();
```

XmlResolver Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlResolver
   extends System.Object
```

Public Properties

```
public abstract void set_Credentials(
   System.Net.ICredentials value
);
```

Public Methods

```
public abstract System.Object GetEntity(
   System.Uri absoluteUri,
   System.String role,
   System.Type ofObjectToReturn
);
public abstract System.Uri ResolveUri(
   System.Uri baseUri,
   System.String relativeUri
);
```

Protected Constructors

```
protected XmlResolver();
```

XmlSignificantWhitespace Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSignificantWhitespace
   extends System.Xml.XmlCharacterData
```

Public Properties

```
public System.String get_LocalName();
public System.String get_Name();
public System.Xml.XmlNodeType get_NodeType();
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public void WriteContentTo(
   System.Xml.XmlWriter w
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

Protected Constructors

```
protected XmlSignificantWhitespace(
   System.String strData,
   System.Xml.XmlDocument doc
);
```

XmlSpace Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlSpace
```

Members

Default

None

Preserve

XmlText Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlText
   extends System.Xml.XmlCharacterData
```

Public Properties

```
public System.String get_LocalName();
public System.String get_Name();
public System.Xml.XmlNodeType get_NodeType();
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public System.Xml.XmlText SplitText(
   int offset
);
public void WriteContentTo(
   System.Xml.XmlWriter w
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

Protected Constructors

```
protected XmlText(
   System.String strData,
   System.Xml.XmlDocument doc
);
```

XmlTextReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class XmlTextReader
   extends System.Xml.XmlReader
   implements System.Xml.IXmlLineInfo
```

Public Constructors

```
public XmlTextReader(
   System.IO.Stream input
);
public XmlTextReader(
   System.IO.TextReader input
);
public XmlTextReader(
   System.String url
);
public XmlTextReader(
   System.IO.Stream input,
   System.Xml.XmlNameTable nt
);
public XmlTextReader(
   System.IO.TextReader input,
   System.Xml.XmlNameTable nt
);
public XmlTextReader(
   System.String url,
   System.IO.Stream input
);
public XmlTextReader(
   System.String url,
   System.IO.TextReader input
);
public XmlTextReader(
   System.String url,
   System.Xml.XmlNameTable nt
);
public XmlTextReader(
   System.IO.Stream xmlFragment,
   System.Xml.XmlNodeType fragType,
   System.Xml.XmlParserContext context
);
public XmlTextReader(
   System.String url,
   System.IO.Stream input,
   System.Xml.XmlNameTable nt
);
public XmlTextReader(
   System.String url,
   System.IO.TextReader input,
   System.Xml.XmlNameTable nt
);
public XmlTextReader(
   System.String xmlFragment,
   System.Xml.XmlNodeType fragType,
   System.Xml.XmlParserContext context
);
```

Public Properties

```
public int get_AttributeCount();
public System.String get_BaseURI();
public int get_Depth();
public System.Text.Encoding get_Encoding();
public boolean get_EOF();
public boolean get_HasValue();
public boolean get_IsDefault();
public boolean get_IsEmptyElement();
public System.String get_Item(
   int i
);
public System.String get_Item(
   System.String name
);
public System.String get_Item(
   System.String name,
   System.String namespaceURI
);
public final int get_LineNumber();
public final int get_LinePosition();
public System.String get_LocalName();
public System.String get_Name();
public boolean get_Namespaces();
public void set_Namespaces(
   boolean value
);
public System.String get_NamespaceURI();
public System.Xml.XmlNameTable get_NameTable();
public System.Xml.XmlNodeType get_NodeType();
public boolean get_Normalization();
public void set_Normalization(
   boolean value
);
public System.String get_Prefix();
public char get_QuoteChar();
public System.Xml.ReadState get_ReadState();
public System.String get_Value();
public System.Xml.WhitespaceHandling get_WhitespaceHandling();
public void set_WhitespaceHandling(
   System.Xml.WhitespaceHandling value
);
public System.String get_XmlLang();
public void set_XmlResolver(
   System.Xml.XmlResolver value
);
public System.Xml.XmlSpace get_XmlSpace();
```

Public Methods

```
public void Close();
public System.String GetAttribute(
   int i
);
public System.String GetAttribute(
   System.String name
);
public System.String GetAttribute(
   System.String localName,
   System.String namespaceURI
);
public System.IO.TextReader GetRemainder();
public System.String LookupNamespace(
   System.String prefix
```

```
);
public boolean MoveToAttribute(
   System.String name
);
public void MoveToAttribute(
   int i
);
public boolean MoveToAttribute(
   System.String localName,
   System.String namespaceURI
);
public boolean MoveToElement();
public boolean MoveToFirstAttribute();
public boolean MoveToNextAttribute();
public boolean Read();
public boolean ReadAttributeValue();
public int ReadBase64(
   ubyte[] array,
   int offset,
   int len
);
public int ReadBinHex(
   ubyte[] array,
   int offset,
   int len
);
public int ReadChars(
   char[] buffer,
   int index,
   int count
);
public System.String ReadInnerXml();
public System.String ReadOuterXml();
public System.String ReadString();
public void ResetState();
public void ResolveEntity();
```

Protected Constructors

```
protected XmlTextReader();
protected XmlTextReader(
   System.Xml.XmlNameTable nt
);
```

XmlTextWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlTextWriter
   extends System.Xml.XmlWriter
```

Public Constructors

```
public XmlTextWriter(
   System.IO.TextWriter w
);
public XmlTextWriter(
   System.IO.Stream w,
   System.Text.Encoding encoding
);
public XmlTextWriter(
   System.String filename,
   System.Text.Encoding encoding
);
```

Public Properties

```
public System.IO.Stream get_BaseStream();
public System.Xml.Formatting get_Formatting();
public void set_Formatting(
   System.Xml.Formatting value
);
public int get_Indentation();
public void set_Indentation(
   int value
);
public char get_IndentChar();
public void set_IndentChar(
   char value
);
public boolean get_Namespaces();
public void set_Namespaces(
   boolean value
);
public char get_QuoteChar();
public void set_QuoteChar(
   char value
);
public System.Xml.WriteState get_WriteState();
public System.String get_XmlLang();
public System.Xml.XmlSpace get_XmlSpace();
```

Public Methods

```
public void Close();
public void Flush();
public System.String LookupPrefix(
   System.String ns
);
public void WriteBase64(
   ubyte[] buffer,
```

```
   int index,
   int count
);
public void WriteBinHex(
   ubyte[] buffer,
   int index,
   int count
);
public void WriteCData(
   System.String text
);
public void WriteCharEntity(
   char ch
);
public void WriteChars(
   char[] buffer,
   int index,
   int count
);
public void WriteComment(
   System.String text
);
public void WriteDocType(
   System.String name,
   System.String pubid,
   System.String sysid,
   System.String subset
);
public void WriteEndAttribute();
public void WriteEndDocument();
public void WriteEndElement();
public void WriteEntityRef(
   System.String name
);
public void WriteFullEndElement();
public void WriteName(
   System.String name
);
public void WriteNmToken(
   System.String name
);
public void WriteProcessingInstruction(
   System.String name,
   System.String text
);
public void WriteQualifiedName(
   System.String localName,
   System.String ns
);
public void WriteRaw(
   System.String data
);
public void WriteRaw(
   char[] buffer,
   int index,
   int count
);
public void WriteStartAttribute(
   System.String prefix,
   System.String localName,
   System.String ns
);
public void WriteStartDocument();
public void WriteStartDocument(
   boolean standalone
);
public void WriteStartElement(
   System.String prefix,
```

```
    System.String localName,
    System.String ns
);
public void WriteString(
    System.String text
);
public void WriteSurrogateCharEntity(
    char lowChar,
    char highChar
);
public void WriteWhitespace(
    System.String ws
);
```

XmlTokenizedType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlTokenizedType
```

Members

CDATA

ENTITIES

ENTITY

ENUMERATION

ID

IDREF

IDREFS

NCName

NMTOKEN

NMTOKENS

None

NOTATION

QName

XmlUrlResolver Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlUrlResolver
   extends System.Xml.XmlResolver
```

Public Constructors

```
public XmlUrlResolver();
```

Public Properties

```
public void set_Credentials(
   System.Net.ICredentials value
);
```

Public Methods

```
public System.Object GetEntity(
   System.Uri absoluteUri,
   System.String role,
   System.Type ofObjectToReturn
);
public System.Uri ResolveUri(
   System.Uri baseUri,
   System.String relativeUri
);
```

XmlValidatingReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class XmlValidatingReader
   extends System.Xml.XmlReader
   implements System.Xml.IXmlLineInfo
```

Public Constructors

```
public XmlValidatingReader(
   System.Xml.XmlReader reader
);
public XmlValidatingReader(
   System.IO.Stream xmlFragment,
   System.Xml.XmlNodeType fragType,
   System.Xml.XmlParserContext context
);
public XmlValidatingReader(
   System.String xmlFragment,
   System.Xml.XmlNodeType fragType,
   System.Xml.XmlParserContext context
);
```

Public Properties

```
public int get_AttributeCount();
public System.String get_BaseURI();
public boolean get_CanResolveEntity();
public int get_Depth();
public System.Text.Encoding get_Encoding();
public System.Xml.EntityHandling get_EntityHandling();
public void set_EntityHandling(
   System.Xml.EntityHandling value
);
public boolean get_EOF();
public boolean get_HasValue();
public boolean get_IsDefault();
public boolean get_IsEmptyElement();
public System.String get_Item(
   int i
);
public System.String get_Item(
   System.String name
);
public System.String get_Item(
   System.String name,
   System.String namespaceURI
);
public System.String get_LocalName();
public System.String get_Name();
public boolean get_Namespaces();
public void set_Namespaces(
   boolean value
);
public System.String get_NamespaceURI();
public System.Xml.XmlNameTable get_NameTable();
public System.Xml.XmlNodeType get_NodeType();
```

```
public System.String get_Prefix();
public char get_QuoteChar();
public System.Xml.XmlReader get_Reader();
public System.Xml.ReadState get_ReadState();
public System.Xml.Schema.XmlSchemaCollection get_Schemas();
public System.Object get_SchemaType();
public System.Xml.ValidationType get_ValidationType();
public void set_ValidationType(
   System.Xml.ValidationType value
);
public System.String get_Value();
public System.String get_XmlLang();
public void set_XmlResolver(
   System.Xml.XmlResolver value
);
public System.Xml.XmlSpace get_XmlSpace();
```

Public Methods

```
public void Close();
public System.String GetAttribute(
   int i
);
public System.String GetAttribute(
   System.String name
);
public System.String GetAttribute(
   System.String localName,
   System.String namespaceURI
);
public System.String LookupNamespace(
   System.String prefix
);
public boolean MoveToAttribute(
   System.String name
);
public void MoveToAttribute(
   int i
);
public boolean MoveToAttribute(
   System.String localName,
   System.String namespaceURI
);
public boolean MoveToElement();
public boolean MoveToFirstAttribute();
public boolean MoveToNextAttribute();
public boolean Read();
public boolean ReadAttributeValue();
public System.String ReadInnerXml();
public System.String ReadOuterXml();
public System.String ReadString();
public System.Object ReadTypedValue();
public void ResolveEntity();
```

Public Events

```
public void add_ValidationEventHandler(
   System.Xml.Schema.ValidationEventHandler value
);
public void remove_ValidationEventHandler(
   System.Xml.Schema.ValidationEventHandler value
);
```

XmlWhitespace Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlWhitespace
   extends System.Xml.XmlCharacterData
```

Public Properties

```
public System.String get_LocalName();
public System.String get_Name();
public System.Xml.XmlNodeType get_NodeType();
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Public Methods

```
public System.Xml.XmlNode CloneNode(
   boolean deep
);
public void WriteContentTo(
   System.Xml.XmlWriter w
);
public void WriteTo(
   System.Xml.XmlWriter w
);
```

Protected Constructors

```
protected XmlWhitespace(
   System.String strData,
   System.Xml.XmlDocument doc
);
```

XmlWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlWriter
   extends System.Object
```

Public Properties

```
public abstract System.Xml.WriteState get_WriteState();
public abstract System.String get_XmlLang();
public abstract System.Xml.XmlSpace get_XmlSpace();
```

Public Methods

```
public abstract void Close();
public abstract void Flush();
public abstract System.String LookupPrefix(
   System.String ns
);
public void WriteAttributes(
   System.Xml.XmlReader reader,
   boolean defattr
);
public void WriteAttributeString(
   System.String localName,
   System.String value
);
public void WriteAttributeString(
   System.String localName,
   System.String ns,
   System.String value
);
public void WriteAttributeString(
   System.String prefix,
   System.String localName,
   System.String ns,
   System.String value
);
public abstract void WriteBase64(
   ubyte[] buffer,
   int index,
   int count
);
public abstract void WriteBinHex(
   ubyte[] buffer,
   int index,
   int count
);
public abstract void WriteCData(
   System.String text
);
public abstract void WriteCharEntity(
   char ch
);
public abstract void WriteChars(
   char[] buffer,
   int index,
   int count
```

```
);
public abstract void WriteComment(
   System.String text
);
public abstract void WriteDocType(
   System.String name,
   System.String pubid,
   System.String sysid,
   System.String subset
);
public void WriteElementString(
   System.String localName,
   System.String value
);
public void WriteElementString(
   System.String localName,
   System.String ns,
   System.String value
);
public abstract void WriteEndAttribute();
public abstract void WriteEndDocument();
public abstract void WriteEndElement();
public abstract void WriteEntityRef(
   System.String name
);
public abstract void WriteFullEndElement();
public abstract void WriteName(
   System.String name
);
public abstract void WriteNmToken(
   System.String name
);
public void WriteNode(
   System.Xml.XmlReader reader,
   boolean defattr
);
public abstract void WriteProcessingInstruction(
   System.String name,
   System.String text
);
public abstract void WriteQualifiedName(
   System.String localName,
   System.String ns
);
public abstract void WriteRaw(
   System.String data
);
public abstract void WriteRaw(
   char[] buffer,
   int index,
   int count
);
public void WriteStartAttribute(
   System.String localName,
   System.String ns
);
public abstract void WriteStartAttribute(
   System.String prefix,
   System.String localName,
   System.String ns
);
public abstract void WriteStartDocument();
public abstract void WriteStartDocument(
   boolean standalone
);
public void WriteStartElement(
   System.String localName
);
```

```
public void WriteStartElement(
   System.String localName,
   System.String ns
);
public abstract void WriteStartElement(
   System.String prefix,
   System.String localName,
   System.String ns
);
public abstract void WriteString(
   System.String text
);
public abstract void WriteSurrogateCharEntity(
   char lowChar,
   char highChar
);
public abstract void WriteWhitespace(
   System.String ws
);
```

Protected Constructors

```
protected XmlWriter();
```

System.Xml.Schema Namespace (Visual J# Syntax)

Classes

ValidationEventArgs

XmlSchema

XmlSchemaAll

XmlSchemaAnnotated

XmlSchemaAnnotation

XmlSchemaAny

XmlSchemaAnyAttribute

XmlSchemaAppInfo

XmlSchemaAttribute

XmlSchemaAttributeGroup

XmlSchemaAttributeGroupRef

XmlSchemaChoice

XmlSchemaCollection

XmlSchemaCollectionEnumerator

XmlSchemaComplexContent

XmlSchemaComplexContentExtension

XmlSchemaComplexContentRestriction

XmlSchemaComplexType

XmlSchemaContent

XmlSchemaContentModel

XmlSchemaDatatype

XmlSchemaDocumentation

XmlSchemaElement

XmlSchemaEnumerationFacet

XmlSchemaException

XmlSchemaExternal

XmlSchemaFacet

XmlSchemaFractionDigitsFacet

XmlSchemaGroup

XmlSchemaGroupBase

XmlSchemaGroupRef

XmlSchemaIdentityConstraint

XmlSchemaImport

XmlSchemaInclude

XmlSchemaKey

XmlSchemaKeyref

XmlSchemaLengthFacet

XmlSchemaMaxExclusiveFacet

XmlSchemaMaxInclusiveFacet

XmlSchemaMaxLengthFacet

XmlSchemaMinExclusiveFacet

XmlSchemaMinInclusiveFacet

XmlSchemaMinLengthFacet

XmlSchemaNotation

XmlSchemaNumericFacet

XmlSchemaObject

XmlSchemaObjectCollection

XmlSchemaObjectEnumerator

XmlSchemaObjectTable

XmlSchemaParticle

XmlSchemaPatternFacet

XmlSchemaRedefine

XmlSchemaSequence

XmlSchemaSimpleContent

XmlSchemaSimpleContentExtension

XmlSchemaSimpleContentRestriction

XmlSchemaSimpleType

XmlSchemaSimpleTypeContent

XmlSchemaSimpleTypeList

XmlSchemaSimpleTypeRestriction

XmlSchemaSimpleTypeUnion

XmlSchemaTotalDigitsFacet

XmlSchemaType

XmlSchemaUnique

XmlSchemaWhiteSpaceFacet

XmlSchemaXPath

Delegates

ValidationEventHandler

Enumerations

XmlSchemaContentProcessing

XmlSchemaContentType

XmlSchemaDerivationMethod

XmlSchemaForm

XmlSchemaUse

XmlSeverityType

ValidationEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class ValidationEventArgs
   extends System.EventArgs
```

Public Properties

```
public System.Xml.Schema.XmlSchemaException get_Exception();
public System.String get_Message();
public System.Xml.Schema.XmlSeverityType get_Severity();
```

ValidationEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void ValidationEventHandler(
   System.Object sender,
   System.Xml.Schema.ValidationEventArgs e
);
```

XmlSchema Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Xml.Serialization.XmlRootAttribute(ElementName = "schema", Namespace =
"http://www.w3.org/2001/XMLSchema", DataType = "", IsNullable = true) */
public class XmlSchema
   extends System.Xml.Schema.XmlSchemaObject
```

Public Constructors

```
public XmlSchema();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaForm get_AttributeFormDefault();
public void set_AttributeFormDefault(
   System.Xml.Schema.XmlSchemaForm value
);
public System.Xml.Schema.XmlSchemaObjectTable get_AttributeGroups();
public System.Xml.Schema.XmlSchemaObjectTable get_Attributes();
public System.Xml.Schema.XmlSchemaDerivationMethod get_BlockDefault();
public void set_BlockDefault(
   System.Xml.Schema.XmlSchemaDerivationMethod value
);
public System.Xml.Schema.XmlSchemaForm get_ElementFormDefault();
public void set_ElementFormDefault(
   System.Xml.Schema.XmlSchemaForm value
);
public System.Xml.Schema.XmlSchemaObjectTable get_Elements();
public System.Xml.Schema.XmlSchemaDerivationMethod get_FinalDefault();
public void set_FinalDefault(
   System.Xml.Schema.XmlSchemaDerivationMethod value
);
public System.Xml.Schema.XmlSchemaObjectTable get_Groups();
public System.String get_Id();
public void set_Id(
   System.String value
);
public System.Xml.Schema.XmlSchemaObjectCollection get_Includes();
public boolean get_IsCompiled();
public System.Xml.Schema.XmlSchemaObjectCollection get_Items();
public System.Xml.Schema.XmlSchemaObjectTable get_Notations();
public System.Xml.Schema.XmlSchemaObjectTable get_SchemaTypes();
public System.String get_TargetNamespace();
public void set_TargetNamespace(
   System.String value
);
public System.Xml.XmlAttribute[] get_UnhandledAttributes();
public void set_UnhandledAttributes(
   System.Xml.XmlAttribute[] value
);
public System.String get_Version();
public void set_Version(
   System.String value
);
```

Public Methods

```
public void Compile(
   System.Xml.Schema.ValidationEventHandler validationEventHandler
);
public static System.Xml.Schema.XmlSchema Read(
   System.IO.Stream stream,
   System.Xml.Schema.ValidationEventHandler validationEventHandler
);
public static System.Xml.Schema.XmlSchema Read(
   System.IO.TextReader reader,
   System.Xml.Schema.ValidationEventHandler validationEventHandler
);
public static System.Xml.Schema.XmlSchema Read(
   System.Xml.XmlReader reader,
   System.Xml.Schema.ValidationEventHandler validationEventHandler
);
public void Write(
   System.IO.Stream stream
);
public void Write(
   System.IO.TextWriter writer
);
public void Write(
   System.Xml.XmlWriter writer
);
public void Write(
   System.IO.Stream stream,
   System.Xml.XmlNamespaceManager namespaceManager
);
public void Write(
   System.IO.TextWriter writer,
   System.Xml.XmlNamespaceManager namespaceManager
);
public void Write(
   System.Xml.XmlWriter writer,
   System.Xml.XmlNamespaceManager namespaceManager
);
```

Public Fields

```
public static System.String InstanceNamespace;
public static System.String Namespace;
```

XmlSchemaAll Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaAll
   extends System.Xml.Schema.XmlSchemaGroupBase
```

Public Constructors

```
public XmlSchemaAll();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaObjectCollection get_Items();
```

XmlSchemaAnnotated Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaAnnotated
   extends System.Xml.Schema.XmlSchemaObject
```

Public Constructors

```
public XmlSchemaAnnotated();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaAnnotation get_Annotation();
public void set_Annotation(
   System.Xml.Schema.XmlSchemaAnnotation value
);
public System.String get_Id();
public void set_Id(
   System.String value
);
public System.Xml.XmlAttribute[] get_UnhandledAttributes();
public void set_UnhandledAttributes(
   System.Xml.XmlAttribute[] value
);
```

XmlSchemaAnnotation Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaAnnotation
   extends System.Xml.Schema.XmlSchemaObject
```

Public Constructors

```
public XmlSchemaAnnotation();
```

Public Properties

```
public System.String get_Id();
public void set_Id(
   System.String value
);
public System.Xml.Schema.XmlSchemaObjectCollection get_Items();
public System.Xml.XmlAttribute[] get_UnhandledAttributes();
public void set_UnhandledAttributes(
   System.Xml.XmlAttribute[] value
);
```

XmlSchemaAny Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaAny
   extends System.Xml.Schema.XmlSchemaParticle
```

Public Constructors

```
public XmlSchemaAny();
```

Public Properties

```
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.Xml.Schema.XmlSchemaContentProcessing get_ProcessContents();
public void set_ProcessContents(
   System.Xml.Schema.XmlSchemaContentProcessing value
);
```

XmlSchemaAnyAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaAnyAttribute
   extends System.Xml.Schema.XmlSchemaAnnotated
```

Public Constructors

```
public XmlSchemaAnyAttribute();
```

Public Properties

```
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.Xml.Schema.XmlSchemaContentProcessing get_ProcessContents();
public void set_ProcessContents(
   System.Xml.Schema.XmlSchemaContentProcessing value
);
```

XmlSchemaAppInfo Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaAppInfo
   extends System.Xml.Schema.XmlSchemaObject
```

Public Constructors

```
public XmlSchemaAppInfo();
```

Public Properties

```
public System.Xml.XmlNode[] get_Markup();
public void set_Markup(
   System.Xml.XmlNode[] value
);
public System.String get_Source();
public void set_Source(
   System.String value
);
```

XmlSchemaAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaAttribute
   extends System.Xml.Schema.XmlSchemaAnnotated
```

Public Constructors

```
public XmlSchemaAttribute();
```

Public Properties

```
public System.Object get_AttributeType();
public System.String get_DefaultValue();
public void set_DefaultValue(
   System.String value
);
public System.String get_FixedValue();
public void set_FixedValue(
   System.String value
);
public System.Xml.Schema.XmlSchemaForm get_Form();
public void set_Form(
   System.Xml.Schema.XmlSchemaForm value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Xml.XmlQualifiedName get_QualifiedName();
public System.Xml.XmlQualifiedName get_RefName();
public void set_RefName(
   System.Xml.XmlQualifiedName value
);
public System.Xml.Schema.XmlSchemaSimpleType get_SchemaType();
public void set_SchemaType(
   System.Xml.Schema.XmlSchemaSimpleType value
);
public System.Xml.XmlQualifiedName get_SchemaTypeName();
public void set_SchemaTypeName(
   System.Xml.XmlQualifiedName value
);
public System.Xml.Schema.XmlSchemaUse get_Use();
public void set_Use(
   System.Xml.Schema.XmlSchemaUse value
);
```

XmlSchemaAttributeGroup Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaAttributeGroup
   extends System.Xml.Schema.XmlSchemaAnnotated
```

Public Constructors

```
public XmlSchemaAttributeGroup();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaAnyAttribute get_AnyAttribute();
public void set_AnyAttribute(
   System.Xml.Schema.XmlSchemaAnyAttribute value
);
public System.Xml.Schema.XmlSchemaObjectCollection get_Attributes();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Xml.Schema.XmlSchemaAttributeGroup get_RedefinedAttributeGroup();
```

XmlSchemaAttributeGroupRef Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaAttributeGroupRef
   extends System.Xml.Schema.XmlSchemaAnnotated
```

Public Constructors

```
public XmlSchemaAttributeGroupRef();
```

Public Properties

```
public System.Xml.XmlQualifiedName get_RefName();
public void set_RefName(
   System.Xml.XmlQualifiedName value
);
```

XmlSchemaChoice Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaChoice
   extends System.Xml.Schema.XmlSchemaGroupBase
```

Public Constructors

```
public XmlSchemaChoice();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaObjectCollection get_Items();
```

XmlSchemaCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class XmlSchemaCollection
   extends System.Object
   implements System.Collections.ICollection, System.Collections.IEnumerable
```

Public Constructors

```
public XmlSchemaCollection();
public XmlSchemaCollection(
   System.Xml.XmlNameTable nametable
);
```

Public Properties

```
public int get_Count();
public System.Xml.Schema.XmlSchema get_Item(
   System.String ns
);
public System.Xml.XmlNameTable get_NameTable();
```

Public Methods

```
public System.Xml.Schema.XmlSchema Add(
   System.Xml.Schema.XmlSchema schema
);
public void Add(
   System.Xml.Schema.XmlSchemaCollection schema
);
public System.Xml.Schema.XmlSchema Add(
   System.String ns,
   System.String uri
);
public System.Xml.Schema.XmlSchema Add(
   System.String ns,
   System.Xml.XmlReader reader
);
public boolean Contains(
   System.String ns
);
public boolean Contains(
   System.Xml.Schema.XmlSchema schema
);
public void CopyTo(
   System.Xml.Schema.XmlSchema[] array,
   int index
);
public System.Xml.Schema.XmlSchemaCollectionEnumerator GetEnumerator();
```

Public Events

```
public void add_ValidationEventHandler(
```

```
    System.Xml.Schema.ValidationEventHandler value
);
public void remove_ValidationEventHandler(
    System.Xml.Schema.ValidationEventHandler value
);
```

XmlSchemaCollectionEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaCollectionEnumerator
   extends System.Object
   implements System.Collections.IEnumerator
```

Public Properties

```
public System.Xml.Schema.XmlSchema get_Current();
```

Public Methods

```
public boolean MoveNext();
```

XmlSchemaComplexContent Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaComplexContent
   extends System.Xml.Schema.XmlSchemaContentModel
```

Public Constructors

```
public XmlSchemaComplexContent();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaContent get_Content();
public void set_Content(
   System.Xml.Schema.XmlSchemaContent value
);
public boolean get_IsMixed();
public void set_IsMixed(
   boolean value
);
```

XmlSchemaComplexContentExtension Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaComplexContentExtension
   extends System.Xml.Schema.XmlSchemaContent
```

Public Constructors

```
public XmlSchemaComplexContentExtension();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaAnyAttribute get_AnyAttribute();
public void set_AnyAttribute(
   System.Xml.Schema.XmlSchemaAnyAttribute value
);
public System.Xml.Schema.XmlSchemaObjectCollection get_Attributes();
public System.Xml.XmlQualifiedName get_BaseTypeName();
public void set_BaseTypeName(
   System.Xml.XmlQualifiedName value
);
public System.Xml.Schema.XmlSchemaParticle get_Particle();
public void set_Particle(
   System.Xml.Schema.XmlSchemaParticle value
);
```

XmlSchemaComplexContentRestriction Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaComplexContentRestriction
   extends System.Xml.Schema.XmlSchemaContent
```

Public Constructors

```
public XmlSchemaComplexContentRestriction();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaAnyAttribute get_AnyAttribute();
public void set_AnyAttribute(
   System.Xml.Schema.XmlSchemaAnyAttribute value
);
public System.Xml.Schema.XmlSchemaObjectCollection get_Attributes();
public System.Xml.XmlQualifiedName get_BaseTypeName();
public void set_BaseTypeName(
   System.Xml.XmlQualifiedName value
);
public System.Xml.Schema.XmlSchemaParticle get_Particle();
public void set_Particle(
   System.Xml.Schema.XmlSchemaParticle value
);
```

XmlSchemaComplexType Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaComplexType
   extends System.Xml.Schema.XmlSchemaType
```

Public Constructors

```
public XmlSchemaComplexType();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaAnyAttribute get_AnyAttribute();
public void set_AnyAttribute(
   System.Xml.Schema.XmlSchemaAnyAttribute value
);
public System.Xml.Schema.XmlSchemaObjectCollection get_Attributes();
public System.Xml.Schema.XmlSchemaObjectTable get_AttributeUses();
public System.Xml.Schema.XmlSchemaAnyAttribute get_AttributeWildcard();
public System.Xml.Schema.XmlSchemaDerivationMethod get_Block();
public void set_Block(
   System.Xml.Schema.XmlSchemaDerivationMethod value
);
public System.Xml.Schema.XmlSchemaDerivationMethod get_BlockResolved();
public System.Xml.Schema.XmlSchemaContentModel get_ContentModel();
public void set_ContentModel(
   System.Xml.Schema.XmlSchemaContentModel value
);
public System.Xml.Schema.XmlSchemaContentType get_ContentType();
public System.Xml.Schema.XmlSchemaParticle get_ContentTypeParticle();
public boolean get_IsAbstract();
public void set_IsAbstract(
   boolean value
);
public boolean get_IsMixed();
public void set_IsMixed(
   boolean value
);
public System.Xml.Schema.XmlSchemaParticle get_Particle();
public void set_Particle(
   System.Xml.Schema.XmlSchemaParticle value
);
```

XmlSchemaContent Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlSchemaContent
   extends System.Xml.Schema.XmlSchemaAnnotated
```

Protected Constructors

```
protected XmlSchemaContent();
```

XmlSchemaContentModel Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlSchemaContentModel
   extends System.Xml.Schema.XmlSchemaAnnotated
```

Public Properties

```
public abstract System.Xml.Schema.XmlSchemaContent get_Content();
public abstract void set_Content(
   System.Xml.Schema.XmlSchemaContent value
);
```

Protected Constructors

```
protected XmlSchemaContentModel();
```

XmlSchemaContentProcessing Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlSchemaContentProcessing
```

Members

Lax

None

Skip

Strict

XmlSchemaContentType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlSchemaContentType
```

Members

ElementOnly

Empty

Mixed

TextOnly

XmlSchemaDatatype Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlSchemaDatatype
   extends System.Object
```

Public Properties

```
public abstract System.Xml.XmlTokenizedType get_TokenizedType();
public abstract System.Type get_ValueType();
```

Public Methods

```
public abstract System.Object ParseValue(
   System.String s,
   System.Xml.XmlNameTable nameTable,
   System.Xml.XmlNamespaceManager nsmgr
);
```

Protected Constructors

```
protected XmlSchemaDatatype();
```

XmlSchemaDerivationMethod Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
/** @attribute System.FlagsAttribute() */
public enum XmlSchemaDerivationMethod
```

Members

All

Empty

Extension

List

None

Restriction

Substitution

Union

XmlSchemaDocumentation Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaDocumentation
   extends System.Xml.Schema.XmlSchemaObject
```

Public Constructors

```
public XmlSchemaDocumentation();
```

Public Properties

```
public System.String get_Language();
public void set_Language(
   System.String value
);
public System.Xml.XmlNode[] get_Markup();
public void set_Markup(
   System.Xml.XmlNode[] value
);
public System.String get_Source();
public void set_Source(
   System.String value
);
```

XmlSchemaElement Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaElement
   extends System.Xml.Schema.XmlSchemaParticle
```

Public Constructors

```
public XmlSchemaElement();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaDerivationMethod get_Block();
public void set_Block(
   System.Xml.Schema.XmlSchemaDerivationMethod value
);
public System.Xml.Schema.XmlSchemaDerivationMethod get_BlockResolved();
public System.Xml.Schema.XmlSchemaObjectCollection get_Constraints();
public System.String get_DefaultValue();
public void set_DefaultValue(
   System.String value
);
public System.Object get_ElementType();
public System.Xml.Schema.XmlSchemaDerivationMethod get_Final();
public void set_Final(
   System.Xml.Schema.XmlSchemaDerivationMethod value
);
public System.Xml.Schema.XmlSchemaDerivationMethod get_FinalResolved();
public System.String get_FixedValue();
public void set_FixedValue(
   System.String value
);
public System.Xml.Schema.XmlSchemaForm get_Form();
public void set_Form(
   System.Xml.Schema.XmlSchemaForm value
);
public boolean get_IsAbstract();
public void set_IsAbstract(
   boolean value
);
public boolean get_IsNillable();
public void set_IsNillable(
   boolean value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Xml.XmlQualifiedName get_QualifiedName();
public System.Xml.XmlQualifiedName get_RefName();
public void set_RefName(
   System.Xml.XmlQualifiedName value
);
public System.Xml.Schema.XmlSchemaType get_SchemaType();
public void set_SchemaType(
   System.Xml.Schema.XmlSchemaType value
);
public System.Xml.XmlQualifiedName get_SchemaTypeName();
```

```
public void set_SchemaTypeName(
   System.Xml.XmlQualifiedName value
);
public System.Xml.XmlQualifiedName get_SubstitutionGroup();
public void set_SubstitutionGroup(
   System.Xml.XmlQualifiedName value
);
```

XmlSchemaEnumerationFacet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaEnumerationFacet
   extends System.Xml.Schema.XmlSchemaFacet
```

Public Constructors

```
public XmlSchemaEnumerationFacet();
```

XmlSchemaException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class XmlSchemaException
   extends System.SystemException
```

Public Constructors

```
public XmlSchemaException(
   System.String message,
   System.Exception innerException
);
```

Public Properties

```
public int get_LineNumber();
public int get_LinePosition();
public System.String get_Message();
public System.Xml.Schema.XmlSchemaObject get_SourceSchemaObject();
public System.String get_SourceUri();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected XmlSchemaException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

XmlSchemaExternal Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlSchemaExternal
   extends System.Xml.Schema.XmlSchemaObject
```

Public Properties

```
public System.String get_Id();
public void set_Id(
   System.String value
);
public System.Xml.Schema.XmlSchema get_Schema();
public void set_Schema(
   System.Xml.Schema.XmlSchema value
);
public System.String get_SchemaLocation();
public void set_SchemaLocation(
   System.String value
);
public System.Xml.XmlAttribute[] get_UnhandledAttributes();
public void set_UnhandledAttributes(
   System.Xml.XmlAttribute[] value
);
```

Protected Constructors

```
protected XmlSchemaExternal();
```

XmlSchemaFacet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlSchemaFacet
   extends System.Xml.Schema.XmlSchemaAnnotated
```

Public Properties

```
public boolean get_IsFixed();
public void set_IsFixed(
   boolean value
);
public System.String get_Value();
public void set_Value(
   System.String value
);
```

Protected Constructors

```
protected XmlSchemaFacet();
```

XmlSchemaForm Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlSchemaForm
```

Members

None

Qualified

Unqualified

XmlSchemaFractionDigitsFacet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaFractionDigitsFacet
   extends System.Xml.Schema.XmlSchemaNumericFacet
```

Public Constructors

```
public XmlSchemaFractionDigitsFacet();
```

XmlSchemaGroup Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaGroup
   extends System.Xml.Schema.XmlSchemaAnnotated
```

Public Constructors

```
public XmlSchemaGroup();
```

Public Properties

```
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Xml.Schema.XmlSchemaGroupBase get_Particle();
public void set_Particle(
   System.Xml.Schema.XmlSchemaGroupBase value
);
```

XmlSchemaGroupBase Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlSchemaGroupBase
   extends System.Xml.Schema.XmlSchemaParticle
```

Public Properties

```
public abstract System.Xml.Schema.XmlSchemaObjectCollection get_Items();
```

Protected Constructors

```
protected XmlSchemaGroupBase();
```

XmlSchemaGroupRef Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaGroupRef
   extends System.Xml.Schema.XmlSchemaParticle
```

Public Constructors

```
public XmlSchemaGroupRef();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaGroupBase get_Particle();
public System.Xml.XmlQualifiedName get_RefName();
public void set_RefName(
   System.Xml.XmlQualifiedName value
);
```

XmlSchemaIdentityConstraint Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaIdentityConstraint
   extends System.Xml.Schema.XmlSchemaAnnotated
```

Public Constructors

```
public XmlSchemaIdentityConstraint();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaObjectCollection get_Fields();
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Xml.XmlQualifiedName get_QualifiedName();
public System.Xml.Schema.XmlSchemaXPath get_Selector();
public void set_Selector(
   System.Xml.Schema.XmlSchemaXPath value
);
```

XmlSchemaImport Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaImport
   extends System.Xml.Schema.XmlSchemaExternal
```

Public Constructors

```
public XmlSchemaImport();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaAnnotation get_Annotation();
public void set_Annotation(
   System.Xml.Schema.XmlSchemaAnnotation value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
```

XmlSchemaInclude Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaInclude
   extends System.Xml.Schema.XmlSchemaExternal
```

Public Constructors

```
public XmlSchemaInclude();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaAnnotation get_Annotation();
public void set_Annotation(
   System.Xml.Schema.XmlSchemaAnnotation value
);
```

XmlSchemaKey Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaKey
   extends System.Xml.Schema.XmlSchemaIdentityConstraint
```

Public Constructors

```
public XmlSchemaKey();
```

XmlSchemaKeyref Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaKeyref
   extends System.Xml.Schema.XmlSchemaIdentityConstraint
```

Public Constructors

```
public XmlSchemaKeyref();
```

Public Properties

```
public System.Xml.XmlQualifiedName get_Refer();
public void set_Refer(
   System.Xml.XmlQualifiedName value
);
```

XmlSchemaLengthFacet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaLengthFacet
   extends System.Xml.Schema.XmlSchemaNumericFacet
```

Public Constructors

```
public XmlSchemaLengthFacet();
```

XmlSchemaMaxExclusiveFacet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaMaxExclusiveFacet
   extends System.Xml.Schema.XmlSchemaFacet
```

Public Constructors

```
public XmlSchemaMaxExclusiveFacet();
```

XmlSchemaMaxInclusiveFacet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaMaxInclusiveFacet
   extends System.Xml.Schema.XmlSchemaFacet
```

Public Constructors

```
public XmlSchemaMaxInclusiveFacet();
```

XmlSchemaMaxLengthFacet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaMaxLengthFacet
   extends System.Xml.Schema.XmlSchemaNumericFacet
```

Public Constructors

```
public XmlSchemaMaxLengthFacet();
```

XmlSchemaMinExclusiveFacet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaMinExclusiveFacet
   extends System.Xml.Schema.XmlSchemaFacet
```

Public Constructors

```
public XmlSchemaMinExclusiveFacet();
```

XmlSchemaMinInclusiveFacet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaMinInclusiveFacet
   extends System.Xml.Schema.XmlSchemaFacet
```

Public Constructors

```
public XmlSchemaMinInclusiveFacet();
```

XmlSchemaMinLengthFacet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaMinLengthFacet
   extends System.Xml.Schema.XmlSchemaNumericFacet
```

Public Constructors

```
public XmlSchemaMinLengthFacet();
```

XmlSchemaNotation Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaNotation
   extends System.Xml.Schema.XmlSchemaAnnotated
```

Public Constructors

```
public XmlSchemaNotation();
```

Public Properties

```
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_Public();
public void set_Public(
   System.String value
);
public System.String get_System();
public void set_System(
   System.String value
);
```

XmlSchemaNumericFacet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlSchemaNumericFacet
   extends System.Xml.Schema.XmlSchemaFacet
```

Protected Constructors

```
protected XmlSchemaNumericFacet();
```

XmlSchemaObject Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlSchemaObject
   extends System.Object
```

Public Properties

```
public int get_LineNumber();
public void set_LineNumber(
   int value
);
public int get_LinePosition();
public void set_LinePosition(
   int value
);
public System.Xml.Serialization.XmlSerializerNamespaces get_Namespaces();
public void set_Namespaces(
   System.Xml.Serialization.XmlSerializerNamespaces value
);
public System.String get_SourceUri();
public void set_SourceUri(
   System.String value
);
```

Protected Constructors

```
protected XmlSchemaObject();
```

XmlSchemaObjectCollection Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class XmlSchemaObjectCollection
   extends System.Collections.CollectionBase
```

Public Constructors

```
public XmlSchemaObjectCollection();
public XmlSchemaObjectCollection(
   System.Xml.Schema.XmlSchemaObject parent
);
```

Public Properties

```
public System.Xml.Schema.XmlSchemaObject get_Item(
   int index
);
public void set_Item(
   int index,
   System.Xml.Schema.XmlSchemaObject value
);
```

Public Methods

```
public int Add(
   System.Xml.Schema.XmlSchemaObject item
);
public boolean Contains(
   System.Xml.Schema.XmlSchemaObject item
);
public void CopyTo(
   System.Xml.Schema.XmlSchemaObject[] array,
   int index
);
public System.Xml.Schema.XmlSchemaObjectEnumerator GetEnumerator();
public int IndexOf(
   System.Xml.Schema.XmlSchemaObject item
);
public void Insert(
   int index,
   System.Xml.Schema.XmlSchemaObject item
);
public void Remove(
   System.Xml.Schema.XmlSchemaObject item
);
```

Protected Methods

```
protected void OnClear();
protected void OnInsert(
   int index,
   System.Object item
```

```
);
protected void OnRemove(
   int index,
   System.Object item
);
protected void OnSet(
   int index,
   System.Object oldValue,
   System.Object newValue
);
```

XmlSchemaObjectEnumerator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaObjectEnumerator
   extends System.Object
   implements System.Collections.IEnumerator
```

Public Properties

```
public System.Xml.Schema.XmlSchemaObject get_Current();
```

Public Methods

```
public boolean MoveNext();
public void Reset();
```

XmlSchemaObjectTable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class XmlSchemaObjectTable
   extends System.Object
```

Public Properties

```
public int get_Count();
public System.Xml.Schema.XmlSchemaObject get_Item(
   System.Xml.XmlQualifiedName name
);
public System.Collections.ICollection get_Names();
public System.Collections.ICollection get_Values();
```

Public Methods

```
public boolean Contains(
   System.Xml.XmlQualifiedName name
);
public System.Collections.IDictionaryEnumerator GetEnumerator();
```

XmlSchemaParticle Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlSchemaParticle
   extends System.Xml.Schema.XmlSchemaAnnotated
```

Public Properties

```
public System.Decimal get_MaxOccurs();
public void set_MaxOccurs(
   System.Decimal value
);
public System.String get_MaxOccursString();
public void set_MaxOccursString(
   System.String value
);
public System.Decimal get_MinOccurs();
public void set_MinOccurs(
   System.Decimal value
);
public System.String get_MinOccursString();
public void set_MinOccursString(
   System.String value
);
```

Protected Constructors

```
protected XmlSchemaParticle();
```

XmlSchemaPatternFacet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaPatternFacet
   extends System.Xml.Schema.XmlSchemaFacet
```

Public Constructors

```
public XmlSchemaPatternFacet();
```

XmlSchemaRedefine Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaRedefine
   extends System.Xml.Schema.XmlSchemaExternal
```

Public Constructors

```
public XmlSchemaRedefine();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaObjectTable get_AttributeGroups();
public System.Xml.Schema.XmlSchemaObjectTable get_Groups();
public System.Xml.Schema.XmlSchemaObjectCollection get_Items();
public System.Xml.Schema.XmlSchemaObjectTable get_SchemaTypes();
```

XmlSchemaSequence Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaSequence
   extends System.Xml.Schema.XmlSchemaGroupBase
```

Public Constructors

```
public XmlSchemaSequence();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaObjectCollection get_Items();
```

XmlSchemaSimpleContent Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaSimpleContent
   extends System.Xml.Schema.XmlSchemaContentModel
```

Public Constructors

```
public XmlSchemaSimpleContent();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaContent get_Content();
public void set_Content(
   System.Xml.Schema.XmlSchemaContent value
);
```

XmlSchemaSimpleContentExtension Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaSimpleContentExtension
   extends System.Xml.Schema.XmlSchemaContent
```

Public Constructors

```
public XmlSchemaSimpleContentExtension();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaAnyAttribute get_AnyAttribute();
public void set_AnyAttribute(
   System.Xml.Schema.XmlSchemaAnyAttribute value
);
public System.Xml.Schema.XmlSchemaObjectCollection get_Attributes();
public System.Xml.XmlQualifiedName get_BaseTypeName();
public void set_BaseTypeName(
   System.Xml.XmlQualifiedName value
);
```

XmlSchemaSimpleContentRestriction Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaSimpleContentRestriction
   extends System.Xml.Schema.XmlSchemaContent
```

Public Constructors

```
public XmlSchemaSimpleContentRestriction();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaAnyAttribute get_AnyAttribute();
public void set_AnyAttribute(
   System.Xml.Schema.XmlSchemaAnyAttribute value
);
public System.Xml.Schema.XmlSchemaObjectCollection get_Attributes();
public System.Xml.Schema.XmlSchemaSimpleType get_BaseType();
public void set_BaseType(
   System.Xml.Schema.XmlSchemaSimpleType value
);
public System.Xml.XmlQualifiedName get_BaseTypeName();
public void set_BaseTypeName(
   System.Xml.XmlQualifiedName value
);
public System.Xml.Schema.XmlSchemaObjectCollection get_Facets();
```

XmlSchemaSimpleType Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaSimpleType
   extends System.Xml.Schema.XmlSchemaType
```

Public Constructors

```
public XmlSchemaSimpleType();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaSimpleTypeContent get_Content();
public void set_Content(
   System.Xml.Schema.XmlSchemaSimpleTypeContent value
);
```

XmlSchemaSimpleTypeContent Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlSchemaSimpleTypeContent
   extends System.Xml.Schema.XmlSchemaAnnotated
```

Protected Constructors

```
protected XmlSchemaSimpleTypeContent();
```

XmlSchemaSimpleTypeList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaSimpleTypeList
   extends System.Xml.Schema.XmlSchemaSimpleTypeContent
```

Public Constructors

```
public XmlSchemaSimpleTypeList();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaSimpleType get_ItemType();
public void set_ItemType(
   System.Xml.Schema.XmlSchemaSimpleType value
);
public System.Xml.XmlQualifiedName get_ItemTypeName();
public void set_ItemTypeName(
   System.Xml.XmlQualifiedName value
);
```

XmlSchemaSimpleTypeRestriction Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaSimpleTypeRestriction
   extends System.Xml.Schema.XmlSchemaSimpleTypeContent
```

Public Constructors

```
public XmlSchemaSimpleTypeRestriction();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaSimpleType get_BaseType();
public void set_BaseType(
   System.Xml.Schema.XmlSchemaSimpleType value
);
public System.Xml.XmlQualifiedName get_BaseTypeName();
public void set_BaseTypeName(
   System.Xml.XmlQualifiedName value
);
public System.Xml.Schema.XmlSchemaObjectCollection get_Facets();
```

XmlSchemaSimpleTypeUnion Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaSimpleTypeUnion
   extends System.Xml.Schema.XmlSchemaSimpleTypeContent
```

Public Constructors

```
public XmlSchemaSimpleTypeUnion();
```

Public Properties

```
public System.Xml.Schema.XmlSchemaObjectCollection get_BaseTypes();
public System.Xml.XmlQualifiedName[] get_MemberTypes();
public void set_MemberTypes(
   System.Xml.XmlQualifiedName[] value
);
```

XmlSchemaTotalDigitsFacet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaTotalDigitsFacet
   extends System.Xml.Schema.XmlSchemaNumericFacet
```

Public Constructors

```
public XmlSchemaTotalDigitsFacet();
```

XmlSchemaType Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaType
   extends System.Xml.Schema.XmlSchemaAnnotated
```

Public Constructors

```
public XmlSchemaType();
```

Public Properties

```
public System.Object get_BaseSchemaType();
public System.Xml.Schema.XmlSchemaDatatype get_Datatype();
public System.Xml.Schema.XmlSchemaDerivationMethod get_DerivedBy();
public System.Xml.Schema.XmlSchemaDerivationMethod get_Final();
public void set_Final(
   System.Xml.Schema.XmlSchemaDerivationMethod value
);
public System.Xml.Schema.XmlSchemaDerivationMethod get_FinalResolved();
public boolean get_IsMixed();
public void set_IsMixed(
   boolean value
);
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.Xml.XmlQualifiedName get_QualifiedName();
```

XmlSchemaUnique Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaUnique
   extends System.Xml.Schema.XmlSchemaIdentityConstraint
```

Public Constructors

```
public XmlSchemaUnique();
```

XmlSchemaUse Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlSchemaUse
```

Members

None

Optional

Prohibited

Required

XmlSchemaWhiteSpaceFacet Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaWhiteSpaceFacet
   extends System.Xml.Schema.XmlSchemaFacet
```

Public Constructors

```
public XmlSchemaWhiteSpaceFacet();
```

XmlSchemaXPath Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaXPath
   extends System.Xml.Schema.XmlSchemaAnnotated
```

Public Constructors

```
public XmlSchemaXPath();
```

Public Properties

```
public System.String get_XPath();
public void set_XPath(
   System.String value
);
```

XmlSeverityType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlSeverityType
```

Members

Error

Warning

System.Xml.Serialization Namespace (Visual J# Syntax)

Classes

CodeIdentifier

CodeIdentifiers

SoapAttributeAttribute

SoapAttributeOverrides

SoapAttributes

SoapCodeExporter

SoapElementAttribute

SoapEnumAttribute

SoapIgnoreAttribute

SoapIncludeAttribute

SoapReflectionImporter

SoapSchemaExporter

SoapSchemaImporter

SoapSchemaMember

SoapTypeAttribute

UnreferencedObjectEventArgs

XmlAnyAttributeAttribute

XmlAnyElementAttribute

XmlAnyElementAttributes

XmlArrayAttribute

XmlArrayItemAttribute

XmlArrayItemAttributes

XmlAttributeAttribute

XmlAttributeEventArgs

XmlAttributeOverrides

XmlAttributes

XmlChoiceIdentifierAttribute

XmlCodeExporter

XmlElementAttribute

XmlElementAttributes

XmlElementEventArgs

XmlEnumAttribute

XmlIgnoreAttribute

XmlIncludeAttribute

XmlMapping

XmlMemberMapping

XmlMembersMapping

XmlNamespaceDeclarationsAttribute

XmlNodeEventArgs

XmlReflectionImporter

XmlReflectionMember

XmlRootAttribute

XmlSchemaExporter

XmlSchemaImporter

XmlSchemas

XmlSerializationReader

XmlSerializationReader.CollectionFixup

XmlSerializationReader.Fixup

XmlSerializationWriter

XmlSerializer

XmlSerializerNamespaces

XmlTextAttribute

XmlTypeAttribute

XmlTypeMapping

Interfaces

IXmlSerializable

Delegates

UnreferencedObjectEventHandler

XmlAttributeEventHandler

XmlElementEventHandler

XmlNodeEventHandler

XmlSerializationCollectionFixupCallback

XmlSerializationFixupCallback

XmlSerializationReadCallback

XmlSerializationWriteCallback

CodeIdentifier Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CodeIdentifier
   extends System.Object
```

Public Constructors

```
public CodeIdentifier();
```

Public Methods

```
public static System.String MakeCamel(
   System.String identifier
);
public static System.String MakePascal(
   System.String identifier
);
public static System.String MakeValid(
   System.String identifier
);
```

CodeIdentifiers Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class CodeIdentifiers
   extends System.Object
```

Public Constructors

```
public CodeIdentifiers();
```

Public Properties

```
public boolean get_UseCamelCasing();
public void set_UseCamelCasing(
   boolean value
);
```

Public Methods

```
public void Add(
   System.String identifier,
   System.Object value
);
public void AddReserved(
   System.String identifier
);
public System.String AddUnique(
   System.String identifier,
   System.Object value
);
public void Clear();
public boolean IsInUse(
   System.String identifier
);
public System.String MakeRightCase(
   System.String identifier
);
public System.String MakeUnique(
   System.String identifier
);
public void Remove(
   System.String identifier
);
public void RemoveReserved(
   System.String identifier
);
public System.Object ToArray(
   System.Type type
);
```

IXmlSerializable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IXmlSerializable
```

Public Methods

```
System.Xml.Schema.XmlSchema GetSchema();
void ReadXml(
   System.Xml.XmlReader reader
);
void WriteXml(
   System.Xml.XmlWriter writer
);
```

SoapAttributeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Parameter, ReturnVal
ue, AllowMultiple = false, Inherited = true) */
public class SoapAttributeAttribute
   extends System.Attribute
```

Public Constructors

```
public SoapAttributeAttribute();
public SoapAttributeAttribute(
   System.String attrName
);
```

Public Properties

```
public System.String get_AttributeName();
public void set_AttributeName(
   System.String value
);
public System.String get_DataType();
public void set_DataType(
   System.String value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
```

SoapAttributeOverrides Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class SoapAttributeOverrides
   extends System.Object
```

Public Constructors

```
public SoapAttributeOverrides();
```

Public Properties

```
public System.Xml.Serialization.SoapAttributes get_Item(
   System.Type type
);
public System.Xml.Serialization.SoapAttributes get_Item(
   System.Type type,
   System.String member
);
```

Public Methods

```
public void Add(
   System.Type type,
   System.Xml.Serialization.SoapAttributes attributes
);
public void Add(
   System.Type type,
   System.String member,
   System.Xml.Serialization.SoapAttributes attributes
);
```

SoapAttributes Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapAttributes
   extends System.Object
```

Public Constructors

```
public SoapAttributes();
public SoapAttributes(
   System.Reflection.ICustomAttributeProvider provider
);
```

Public Properties

```
public System.Xml.Serialization.SoapAttributeAttribute get_SoapAttribute();
public void set_SoapAttribute(
   System.Xml.Serialization.SoapAttributeAttribute value
);
public System.Object get_SoapDefaultValue();
public void set_SoapDefaultValue(
   System.Object value
);
public System.Xml.Serialization.SoapElementAttribute get_SoapElement();
public void set_SoapElement(
   System.Xml.Serialization.SoapElementAttribute value
);
public System.Xml.Serialization.SoapEnumAttribute get_SoapEnum();
public void set_SoapEnum(
   System.Xml.Serialization.SoapEnumAttribute value
);
public boolean get_SoapIgnore();
public void set_SoapIgnore(
   boolean value
);
public System.Xml.Serialization.SoapTypeAttribute get_SoapType();
public void set_SoapType(
   System.Xml.Serialization.SoapTypeAttribute value
);
```

SoapCodeExporter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapCodeExporter
   extends System.Object
```

Public Constructors

```
public SoapCodeExporter(
   System.CodeDom.CodeNamespace codeNamespace
);
public SoapCodeExporter(
   System.CodeDom.CodeNamespace codeNamespace,
   System.CodeDom.CodeCompileUnit codeCompileUnit
);
```

Public Properties

```
public System.CodeDom.CodeAttributeDeclarationCollection get_IncludeMetadata();
```

Public Methods

```
public void AddMappingMetadata(
   System.CodeDom.CodeAttributeDeclarationCollection metadata,
   System.Xml.Serialization.XmlMemberMapping member
);
public void AddMappingMetadata(
   System.CodeDom.CodeAttributeDeclarationCollection metadata,
   System.Xml.Serialization.XmlMemberMapping member,
   boolean forceUseMemberName
);
public void ExportMembersMapping(
   System.Xml.Serialization.XmlMembersMapping xmlMembersMapping
);
public void ExportTypeMapping(
   System.Xml.Serialization.XmlTypeMapping xmlTypeMapping
);
```

SoapElementAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Parameter, ReturnVal
ue, AllowMultiple = false, Inherited = true) */
public class SoapElementAttribute
   extends System.Attribute
```

Public Constructors

```
public SoapElementAttribute();
public SoapElementAttribute(
   System.String elementName
);
```

Public Properties

```
public System.String get_DataType();
public void set_DataType(
   System.String value
);
public System.String get_ElementName();
public void set_ElementName(
   System.String value
);
public boolean get_IsNullable();
public void set_IsNullable(
   boolean value
);
```

SoapEnumAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, AllowMultiple = false, Inherit
ed = true) */
public class SoapEnumAttribute
   extends System.Attribute
```

Public Constructors

```
public SoapEnumAttribute();
public SoapEnumAttribute(
   System.String name
);
```

Public Properties

```
public System.String get_Name();
public void set_Name(
   System.String value
);
```

SoapIgnoreAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Parameter, ReturnVal
ue, AllowMultiple = false, Inherited = true) */
public class SoapIgnoreAttribute
   extends System.Attribute
```

Public Constructors

```
public SoapIgnoreAttribute();
```

SoapIncludeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, Method, AllowMultiple
= true, Inherited = true) */
public class SoapIncludeAttribute
   extends System.Attribute
```

Public Constructors

```
public SoapIncludeAttribute(
   System.Type type
);
```

Public Properties

```
public System.Type get_Type();
public void set_Type(
   System.Type value
);
```

SoapReflectionImporter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapReflectionImporter
   extends System.Object
```

Public Constructors

```
public SoapReflectionImporter();
public SoapReflectionImporter(
   System.String defaultNamespace
);
public SoapReflectionImporter(
   System.Xml.Serialization.SoapAttributeOverrides attributeOverrides
);
public SoapReflectionImporter(
   System.Xml.Serialization.SoapAttributeOverrides attributeOverrides,
   System.String defaultNamespace
);
```

Public Methods

```
public System.Xml.Serialization.XmlMembersMapping ImportMembersMapping(
   System.String elementName,
   System.String ns,
   System.Xml.Serialization.XmlReflectionMember[] members
);
public System.Xml.Serialization.XmlMembersMapping ImportMembersMapping(
   System.String elementName,
   System.String ns,
   System.Xml.Serialization.XmlReflectionMember[] members,
   boolean hasWrapperElement,
   boolean writeAccessors
);
public System.Xml.Serialization.XmlMembersMapping ImportMembersMapping(
   System.String elementName,
   System.String ns,
   System.Xml.Serialization.XmlReflectionMember[] members,
   boolean hasWrapperElement,
   boolean writeAccessors,
   boolean validate
);
public System.Xml.Serialization.XmlTypeMapping ImportTypeMapping(
   System.Type type
);
public System.Xml.Serialization.XmlTypeMapping ImportTypeMapping(
   System.Type type,
   System.String defaultNamespace
);
public void IncludeType(
   System.Type type
);
public void IncludeTypes(
   System.Reflection.ICustomAttributeProvider provider
);
```

SoapSchemaExporter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapSchemaExporter
   extends System.Object
```

Public Constructors

```
public SoapSchemaExporter(
   System.Xml.Serialization.XmlSchemas schemas
);
```

Public Methods

```
public void ExportMembersMapping(
   System.Xml.Serialization.XmlMembersMapping xmlMembersMapping
);
public void ExportMembersMapping(
   System.Xml.Serialization.XmlMembersMapping xmlMembersMapping,
   boolean exportEnclosingType
);
public void ExportTypeMapping(
   System.Xml.Serialization.XmlTypeMapping xmlTypeMapping
);
```

SoapSchemaImporter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapSchemaImporter
   extends System.Object
```

Public Constructors

```
public SoapSchemaImporter(
   System.Xml.Serialization.XmlSchemas schemas
);
public SoapSchemaImporter(
   System.Xml.Serialization.XmlSchemas schemas,
   System.Xml.Serialization.CodeIdentifiers typeIdentifiers
);
```

Public Methods

```
public System.Xml.Serialization.XmlTypeMapping ImportDerivedTypeMapping(
   System.Xml.XmlQualifiedName name,
   System.Type baseType,
   boolean baseTypeCanBeIndirect
);
public System.Xml.Serialization.XmlMembersMapping ImportMembersMapping(
   System.String name,
   System.String ns,
   System.Xml.Serialization.SoapSchemaMember member
);
public System.Xml.Serialization.XmlMembersMapping ImportMembersMapping(
   System.String name,
   System.String ns,
   System.Xml.Serialization.SoapSchemaMember[] members
);
public System.Xml.Serialization.XmlMembersMapping ImportMembersMapping(
   System.String name,
   System.String ns,
   System.Xml.Serialization.SoapSchemaMember[] members,
   boolean hasWrapperElement
);
public System.Xml.Serialization.XmlMembersMapping ImportMembersMapping(
   System.String name,
   System.String ns,
   System.Xml.Serialization.SoapSchemaMember[] members,
   boolean hasWrapperElement,
   System.Type baseType,
   boolean baseTypeCanBeIndirect
);
```

SoapSchemaMember Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class SoapSchemaMember
   extends System.Object
```

Public Constructors

```
public SoapSchemaMember();
```

Public Properties

```
public System.String get_MemberName();
public void set_MemberName(
   System.String value
);
public System.Xml.XmlQualifiedName get_MemberType();
public void set_MemberType(
   System.Xml.XmlQualifiedName value
);
```

SoapTypeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, Enum, Interface, Allow
Multiple = false, Inherited = true) */
public class SoapTypeAttribute
   extends System.Attribute
```

Public Constructors

```
public SoapTypeAttribute();
public SoapTypeAttribute(
   System.String typeName
);
public SoapTypeAttribute(
   System.String typeName,
   System.String ns
);
```

Public Properties

```
public boolean get_IncludeInSchema();
public void set_IncludeInSchema(
   boolean value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.String get_TypeName();
public void set_TypeName(
   System.String value
);
```

UnreferencedObjectEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class UnreferencedObjectEventArgs
   extends System.EventArgs
```

Public Constructors

```
public UnreferencedObjectEventArgs(
   System.Object o,
   System.String id
);
```

Public Properties

```
public System.String get_UnreferencedId();
public System.Object get_UnreferencedObject();
```

UnreferencedObjectEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void UnreferencedObjectEventHandler(
   System.Object sender,
   System.Xml.Serialization.UnreferencedObjectEventArgs e
);
```

XmlAnyAttributeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Parameter, ReturnVal
ue, AllowMultiple = false, Inherited = true) */
public class XmlAnyAttributeAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlAnyAttributeAttribute();
```

XmlAnyElementAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Parameter, ReturnVal
ue, AllowMultiple = true, Inherited = true) */
public class XmlAnyElementAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlAnyElementAttribute();
public XmlAnyElementAttribute(
   System.String name
);
public XmlAnyElementAttribute(
   System.String name,
   System.String ns
);
```

Public Properties

```
public System.String get_Name();
public void set_Name(
   System.String value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
```

XmlAnyElementAttributes Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class XmlAnyElementAttributes
   extends System.Collections.CollectionBase
```

Public Constructors

```
public XmlAnyElementAttributes();
```

Public Properties

```
public System.Xml.Serialization.XmlAnyElementAttribute get_Item(
   int index
);
public void set_Item(
   int index,
   System.Xml.Serialization.XmlAnyElementAttribute value
);
```

Public Methods

```
public int Add(
   System.Xml.Serialization.XmlAnyElementAttribute attribute
);
public boolean Contains(
   System.Xml.Serialization.XmlAnyElementAttribute attribute
);
public void CopyTo(
   System.Xml.Serialization.XmlAnyElementAttribute[] array,
   int index
);
public int IndexOf(
   System.Xml.Serialization.XmlAnyElementAttribute attribute
);
public void Insert(
   int index,
   System.Xml.Serialization.XmlAnyElementAttribute attribute
);
public void Remove(
   System.Xml.Serialization.XmlAnyElementAttribute attribute
);
```

XmlArrayAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Parameter, ReturnVal
ue, AllowMultiple = false, Inherited = true) */
public class XmlArrayAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlArrayAttribute();
public XmlArrayAttribute(
   System.String elementName
);
```

Public Properties

```
public System.String get_ElementName();
public void set_ElementName(
   System.String value
);
public System.Xml.Schema.XmlSchemaForm get_Form();
public void set_Form(
   System.Xml.Schema.XmlSchemaForm value
);
public boolean get_IsNullable();
public void set_IsNullable(
   boolean value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
```

XmlArrayItemAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Parameter, ReturnVal
ue, AllowMultiple = true, Inherited = true) */
public class XmlArrayItemAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlArrayItemAttribute();
public XmlArrayItemAttribute(
   System.String elementName
);
public XmlArrayItemAttribute(
   System.Type type
);
public XmlArrayItemAttribute(
   System.String elementName,
   System.Type type
);
```

Public Properties

```
public System.String get_DataType();
public void set_DataType(
   System.String value
);
public System.String get_ElementName();
public void set_ElementName(
   System.String value
);
public System.Xml.Schema.XmlSchemaForm get_Form();
public void set_Form(
   System.Xml.Schema.XmlSchemaForm value
);
public boolean get_IsNullable();
public void set_IsNullable(
   boolean value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public int get_NestingLevel();
public void set_NestingLevel(
   int value
);
public System.Type get_Type();
public void set_Type(
   System.Type value
);
```

XmlArrayItemAttributes Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class XmlArrayItemAttributes
   extends System.Collections.CollectionBase
```

Public Constructors

```
public XmlArrayItemAttributes();
```

Public Properties

```
public System.Xml.Serialization.XmlArrayItemAttribute get_Item(
   int index
);
public void set_Item(
   int index,
   System.Xml.Serialization.XmlArrayItemAttribute value
);
```

Public Methods

```
public int Add(
   System.Xml.Serialization.XmlArrayItemAttribute attribute
);
public boolean Contains(
   System.Xml.Serialization.XmlArrayItemAttribute attribute
);
public void CopyTo(
   System.Xml.Serialization.XmlArrayItemAttribute[] array,
   int index
);
public int IndexOf(
   System.Xml.Serialization.XmlArrayItemAttribute attribute
);
public void Insert(
   int index,
   System.Xml.Serialization.XmlArrayItemAttribute attribute
);
public void Remove(
   System.Xml.Serialization.XmlArrayItemAttribute attribute
);
```

XmlAttributeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Parameter, ReturnVal
ue, AllowMultiple = false, Inherited = true) */
public class XmlAttributeAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlAttributeAttribute();
public XmlAttributeAttribute(
   System.String attributeName
);
public XmlAttributeAttribute(
   System.Type type
);
public XmlAttributeAttribute(
   System.String attributeName,
   System.Type type
);
```

Public Properties

```
public System.String get_AttributeName();
public void set_AttributeName(
   System.String value
);
public System.String get_DataType();
public void set_DataType(
   System.String value
);
public System.Xml.Schema.XmlSchemaForm get_Form();
public void set_Form(
   System.Xml.Schema.XmlSchemaForm value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.Type get_Type();
public void set_Type(
   System.Type value
);
```

XmlAttributeEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlAttributeEventArgs
   extends System.EventArgs
```

Public Properties

```
public System.Xml.XmlAttribute get_Attr();
public int get_LineNumber();
public int get_LinePosition();
public System.Object get_ObjectBeingDeserialized();
```

XmlAttributeEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void XmlAttributeEventHandler(
   System.Object sender,
   System.Xml.Serialization.XmlAttributeEventArgs e
);
```

XmlAttributeOverrides Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class XmlAttributeOverrides
   extends System.Object
```

Public Constructors

```
public XmlAttributeOverrides();
```

Public Properties

```
public System.Xml.Serialization.XmlAttributes get_Item(
   System.Type type
);
public System.Xml.Serialization.XmlAttributes get_Item(
   System.Type type,
   System.String member
);
```

Public Methods

```
public void Add(
   System.Type type,
   System.Xml.Serialization.XmlAttributes attributes
);
public void Add(
   System.Type type,
   System.String member,
   System.Xml.Serialization.XmlAttributes attributes
);
```

XmlAttributes Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlAttributes
   extends System.Object
```

Public Constructors

```
public XmlAttributes();
public XmlAttributes(
   System.Reflection.ICustomAttributeProvider provider
);
```

Public Properties

```
public System.Xml.Serialization.XmlAnyAttributeAttribute get_XmlAnyAttribute();
public void set_XmlAnyAttribute(
   System.Xml.Serialization.XmlAnyAttributeAttribute value
);
public System.Xml.Serialization.XmlAnyElementAttributes get_XmlAnyElements();
public System.Xml.Serialization.XmlArrayAttribute get_XmlArray();
public void set_XmlArray(
   System.Xml.Serialization.XmlArrayAttribute value
);
public System.Xml.Serialization.XmlArrayItemAttributes get_XmlArrayItems();
public System.Xml.Serialization.XmlAttributeAttribute get_XmlAttribute();
public void set_XmlAttribute(
   System.Xml.Serialization.XmlAttributeAttribute value
);
public System.Xml.Serialization.XmlChoiceIdentifierAttribute get_XmlChoiceIdentifier();
public System.Object get_XmlDefaultValue();
public void set_XmlDefaultValue(
   System.Object value
);
public System.Xml.Serialization.XmlElementAttributes get_XmlElements();
public System.Xml.Serialization.XmlEnumAttribute get_XmlEnum();
public void set_XmlEnum(
   System.Xml.Serialization.XmlEnumAttribute value
);
public boolean get_XmlIgnore();
public void set_XmlIgnore(
   boolean value
);
public boolean get_Xmlns();
public void set_Xmlns(
   boolean value
);
public System.Xml.Serialization.XmlRootAttribute get_XmlRoot();
public void set_XmlRoot(
   System.Xml.Serialization.XmlRootAttribute value
);
public System.Xml.Serialization.XmlTextAttribute get_XmlText();
public void set_XmlText(
   System.Xml.Serialization.XmlTextAttribute value
);
public System.Xml.Serialization.XmlTypeAttribute get_XmlType();
public void set_XmlType(
   System.Xml.Serialization.XmlTypeAttribute value
```

```
);
```

XmlChoiceIdentifierAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Parameter, ReturnVal
ue, AllowMultiple = false, Inherited = true) */
public class XmlChoiceIdentifierAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlChoiceIdentifierAttribute();
public XmlChoiceIdentifierAttribute(
   System.String name
);
```

Public Properties

```
public System.String get_MemberName();
public void set_MemberName(
   System.String value
);
```

XmlCodeExporter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlCodeExporter
   extends System.Object
```

Public Constructors

```
public XmlCodeExporter(
   System.CodeDom.CodeNamespace codeNamespace
);
public XmlCodeExporter(
   System.CodeDom.CodeNamespace codeNamespace,
   System.CodeDom.CodeCompileUnit codeCompileUnit
);
```

Public Properties

```
public System.CodeDom.CodeAttributeDeclarationCollection get_IncludeMetadata();
```

Public Methods

```
public void AddMappingMetadata(
   System.CodeDom.CodeAttributeDeclarationCollection metadata,
   System.Xml.Serialization.XmlMemberMapping member,
   System.String ns
);
public void AddMappingMetadata(
   System.CodeDom.CodeAttributeDeclarationCollection metadata,
   System.Xml.Serialization.XmlTypeMapping mapping,
   System.String ns
);
public void AddMappingMetadata(
   System.CodeDom.CodeAttributeDeclarationCollection metadata,
   System.Xml.Serialization.XmlMemberMapping member,
   System.String ns,
   boolean forceUseMemberName
);
public void ExportMembersMapping(
   System.Xml.Serialization.XmlMembersMapping xmlMembersMapping
);
public void ExportTypeMapping(
   System.Xml.Serialization.XmlTypeMapping xmlTypeMapping
);
```

XmlElementAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Parameter, ReturnVal
ue, AllowMultiple = true, Inherited = true) */
public class XmlElementAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlElementAttribute();
public XmlElementAttribute(
   System.String elementName
);
public XmlElementAttribute(
   System.Type type
);
public XmlElementAttribute(
   System.String elementName,
   System.Type type
);
```

Public Properties

```
public System.String get_DataType();
public void set_DataType(
   System.String value
);
public System.String get_ElementName();
public void set_ElementName(
   System.String value
);
public System.Xml.Schema.XmlSchemaForm get_Form();
public void set_Form(
   System.Xml.Schema.XmlSchemaForm value
);
public boolean get_IsNullable();
public void set_IsNullable(
   boolean value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.Type get_Type();
public void set_Type(
   System.Type value
);
```

XmlElementAttributes Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class XmlElementAttributes
   extends System.Collections.CollectionBase
```

Public Constructors

```
public XmlElementAttributes();
```

Public Properties

```
public System.Xml.Serialization.XmlElementAttribute get_Item(
   int index
);
public void set_Item(
   int index,
   System.Xml.Serialization.XmlElementAttribute value
);
```

Public Methods

```
public int Add(
   System.Xml.Serialization.XmlElementAttribute attribute
);
public boolean Contains(
   System.Xml.Serialization.XmlElementAttribute attribute
);
public void CopyTo(
   System.Xml.Serialization.XmlElementAttribute[] array,
   int index
);
public int IndexOf(
   System.Xml.Serialization.XmlElementAttribute attribute
);
public void Insert(
   int index,
   System.Xml.Serialization.XmlElementAttribute attribute
);
public void Remove(
   System.Xml.Serialization.XmlElementAttribute attribute
);
```

XmlElementEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlElementEventArgs
   extends System.EventArgs
```

Public Properties

```
public System.Xml.XmlElement get_Element();
public int get_LineNumber();
public int get_LinePosition();
public System.Object get_ObjectBeingDeserialized();
```

XmlElementEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void XmlElementEventHandler(
   System.Object sender,
   System.Xml.Serialization.XmlElementEventArgs e
);
```

XmlEnumAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Field, AllowMultiple = false, Inherit
ed = true) */
public class XmlEnumAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlEnumAttribute();
public XmlEnumAttribute(
   System.String name
);
```

Public Properties

```
public System.String get_Name();
public void set_Name(
   System.String value
);
```

XmlIgnoreAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Parameter, ReturnVal
ue, AllowMultiple = false, Inherited = true) */
public class XmlIgnoreAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlIgnoreAttribute();
```

XmlIncludeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, Method, AllowMultiple
= true, Inherited = true) */
public class XmlIncludeAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlIncludeAttribute(
   System.Type type
);
```

Public Properties

```
public System.Type get_Type();
public void set_Type(
   System.Type value
);
```

XmlMapping Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlMapping
   extends System.Object
```

XmlMemberMapping Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlMemberMapping
   extends System.Object
```

Public Properties

```
public boolean get_Any();
public System.String get_ElementName();
public System.String get_MemberName();
public System.String get_Namespace();
public System.String get_TypeFullName();
public System.String get_TypeName();
public System.String get_TypeNamespace();
```

XmlMembersMapping Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class XmlMembersMapping
   extends System.Xml.Serialization.XmlMapping
```

Public Properties

```
public int get_Count();
public System.String get_ElementName();
public System.Xml.Serialization.XmlMemberMapping get_Item(
   int index
);
public System.String get_Namespace();
public System.String get_TypeName();
public System.String get_TypeNamespace();
```

XmlNamespaceDeclarationsAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Parameter, ReturnVal
ue, AllowMultiple = false, Inherited = true) */
public class XmlNamespaceDeclarationsAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlNamespaceDeclarationsAttribute();
```

XmlNodeEventArgs Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlNodeEventArgs
   extends System.EventArgs
```

Public Properties

```
public int get_LineNumber();
public int get_LinePosition();
public System.String get_LocalName();
public System.String get_Name();
public System.String get_NamespaceURI();
public System.Xml.XmlNodeType get_NodeType();
public System.Object get_ObjectBeingDeserialized();
public System.String get_Text();
```

XmlNodeEventHandler Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void XmlNodeEventHandler(
   System.Object sender,
   System.Xml.Serialization.XmlNodeEventArgs e
);
```

XmlReflectionImporter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlReflectionImporter
   extends System.Object
```

Public Constructors

```
public XmlReflectionImporter();
public XmlReflectionImporter(
   System.String defaultNamespace
);
public XmlReflectionImporter(
   System.Xml.Serialization.XmlAttributeOverrides attributeOverrides
);
public XmlReflectionImporter(
   System.Xml.Serialization.XmlAttributeOverrides attributeOverrides,
   System.String defaultNamespace
);
```

Public Methods

```
public System.Xml.Serialization.XmlMembersMapping ImportMembersMapping(
   System.String elementName,
   System.String ns,
   System.Xml.Serialization.XmlReflectionMember[] members,
   boolean hasWrapperElement
);
public System.Xml.Serialization.XmlTypeMapping ImportTypeMapping(
   System.Type type
);
public System.Xml.Serialization.XmlTypeMapping ImportTypeMapping(
   System.Type type,
   System.String defaultNamespace
);
public System.Xml.Serialization.XmlTypeMapping ImportTypeMapping(
   System.Type type,
   System.Xml.Serialization.XmlRootAttribute root
);
public System.Xml.Serialization.XmlTypeMapping ImportTypeMapping(
   System.Type type,
   System.Xml.Serialization.XmlRootAttribute root,
   System.String defaultNamespace
);
public void IncludeType(
   System.Type type
);
public void IncludeTypes(
   System.Reflection.ICustomAttributeProvider provider
);
```

XmlReflectionMember Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlReflectionMember
   extends System.Object
```

Public Constructors

```
public XmlReflectionMember();
```

Public Properties

```
public boolean get_IsReturnValue();
public void set_IsReturnValue(
   boolean value
);
public System.String get_MemberName();
public void set_MemberName(
   System.String value
);
public System.Type get_MemberType();
public void set_MemberType(
   System.Type value
);
public boolean get_OverrideIsNullable();
public void set_OverrideIsNullable(
   boolean value
);
public System.Xml.Serialization.SoapAttributes get_SoapAttributes();
public void set_SoapAttributes(
   System.Xml.Serialization.SoapAttributes value
);
public System.Xml.Serialization.XmlAttributes get_XmlAttributes();
public void set_XmlAttributes(
   System.Xml.Serialization.XmlAttributes value
);
```

XmlRootAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, Enum, Interface, Retur
nValue, AllowMultiple = false, Inherited = true) */
public class XmlRootAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlRootAttribute();
public XmlRootAttribute(
   System.String elementName
);
```

Public Properties

```
public System.String get_DataType();
public void set_DataType(
   System.String value
);
public System.String get_ElementName();
public void set_ElementName(
   System.String value
);
public boolean get_IsNullable();
public void set_IsNullable(
   boolean value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
```

XmlSchemaExporter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaExporter
   extends System.Object
```

Public Constructors

```
public XmlSchemaExporter(
   System.Xml.Serialization.XmlSchemas schemas
);
```

Public Methods

```
public System.String ExportAnyType(
   System.String ns
);
public void ExportMembersMapping(
   System.Xml.Serialization.XmlMembersMapping xmlMembersMapping
);
public System.Xml.XmlQualifiedName ExportTypeMapping(
   System.Xml.Serialization.XmlMembersMapping xmlMembersMapping
);
public void ExportTypeMapping(
   System.Xml.Serialization.XmlTypeMapping xmlTypeMapping
);
```

XmlSchemaImporter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSchemaImporter
   extends System.Object
```

Public Constructors

```
public XmlSchemaImporter(
   System.Xml.Serialization.XmlSchemas schemas
);
public XmlSchemaImporter(
   System.Xml.Serialization.XmlSchemas schemas,
   System.Xml.Serialization.CodeIdentifiers typeIdentifiers
);
```

Public Methods

```
public System.Xml.Serialization.XmlMembersMapping ImportAnyType(
   System.Xml.XmlQualifiedName typeName,
   System.String elementName
);
public System.Xml.Serialization.XmlTypeMapping ImportDerivedTypeMapping(
   System.Xml.XmlQualifiedName name,
   System.Type baseType
);
public System.Xml.Serialization.XmlTypeMapping ImportDerivedTypeMapping(
   System.Xml.XmlQualifiedName name,
   System.Type baseType,
   boolean baseTypeCanBeIndirect
);
public System.Xml.Serialization.XmlMembersMapping ImportMembersMapping(
   System.Xml.XmlQualifiedName name
);
public System.Xml.Serialization.XmlMembersMapping ImportMembersMapping(
   System.Xml.XmlQualifiedName[] names
);
public System.Xml.Serialization.XmlMembersMapping ImportMembersMapping(
   System.Xml.XmlQualifiedName[] names,
   System.Type baseType,
   boolean baseTypeCanBeIndirect
);
public System.Xml.Serialization.XmlTypeMapping ImportTypeMapping(
   System.Xml.XmlQualifiedName name
);
```

XmlSchemas Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.Reflection.DefaultMemberAttribute(MemberName = "Item") */
public class XmlSchemas
   extends System.Collections.CollectionBase
```

Public Constructors

```
public XmlSchemas();
```

Public Properties

```
public System.Xml.Schema.XmlSchema get_Item(
   int index
);
public System.Xml.Schema.XmlSchema get_Item(
   System.String ns
);
public void set_Item(
   int index,
   System.Xml.Schema.XmlSchema value
);
```

Public Methods

```
public int Add(
   System.Xml.Schema.XmlSchema schema
);
public void Add(
   System.Xml.Serialization.XmlSchemas schemas
);
public boolean Contains(
   System.Xml.Schema.XmlSchema schema
);
public void CopyTo(
   System.Xml.Schema.XmlSchema[] array,
   int index
);
public System.Object Find(
   System.Xml.XmlQualifiedName name,
   System.Type type
);
public int IndexOf(
   System.Xml.Schema.XmlSchema schema
);
public void Insert(
   int index,
   System.Xml.Schema.XmlSchema schema
);
public static boolean IsDataSet(
   System.Xml.Schema.XmlSchema schema
);
public void Remove(
   System.Xml.Schema.XmlSchema schema
);
```

Protected Methods

```
protected void OnClear();
protected void OnInsert(
   int index,
   System.Object value
);
protected void OnRemove(
   int index,
   System.Object value
);
protected void OnSet(
   int index,
   System.Object oldValue,
   System.Object newValue
);
```

XmlSerializationCollectionFixupCallback Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void XmlSerializationCollectionFixupCallback(
   System.Object collection,
   System.Object collectionItems
);
```

XmlSerializationFixupCallback Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void XmlSerializationFixupCallback(
   System.Object fixup
);
```

XmlSerializationReadCallback Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
System.Object XmlSerializationReadCallback();
```

XmlSerializationReader Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlSerializationReader
   extends System.Object
```

Protected Constructors

```
protected XmlSerializationReader();
```

Protected Properties

```
protected System.Xml.XmlDocument get_Document();
protected System.Xml.XmlReader get_Reader();
```

Protected Methods

```
protected void AddFixup(
   System.Xml.Serialization.XmlSerializationReader+CollectionFixup fixup
);
protected void AddFixup(
   System.Xml.Serialization.XmlSerializationReader+Fixup fixup
);
protected void AddReadCallback(
   System.String name,
   System.String ns,
   System.Type type,
   System.Xml.Serialization.XmlSerializationReadCallback read
);
protected void AddTarget(
   System.String id,
   System.Object o
);
protected System.Exception CreateAbstractTypeException(
   System.String name,
   System.String ns
);
protected System.Exception CreateInvalidCastException(
   System.Type type,
   System.Object value
);
protected System.Exception CreateReadOnlyCollectionException(
   System.String name
);
protected System.Exception CreateUnknownConstantException(
   System.String value,
   System.Type enumType
);
protected System.Exception CreateUnknownNodeException();
protected System.Exception CreateUnknownTypeException(
   System.Xml.XmlQualifiedName type
);
protected System.Array EnsureArrayIndex(
   System.Array a,
   int index,
   System.Type elementType
```

```
);
protected void FixupArrayRefs(
   System.Object fixup
);
protected int GetArrayLength(
   System.String name,
   System.String ns
);
protected boolean GetNullAttr();
protected System.Object GetTarget(
   System.String id
);
protected System.Xml.XmlQualifiedName GetXsiType();
protected abstract void InitCallbacks();
protected abstract void InitIDs();
protected boolean IsXmlnsAttribute(
   System.String name
);
protected void ParseWsdlArrayType(
   System.Xml.XmlAttribute attr
);
protected System.Xml.XmlQualifiedName ReadElementQualifiedName();
protected void ReadEndElement();
protected boolean ReadNull();
protected System.Xml.XmlQualifiedName ReadNullableQualifiedName();
protected System.String ReadNullableString();
protected boolean ReadReference(
   System.String& fixupReference
);
protected System.Object ReadReferencedElement();
protected System.Object ReadReferencedElement(
   System.String name,
   System.String ns
);
protected void ReadReferencedElements();
protected System.Object ReadReferencingElement(
   System.String& fixupReference
);
protected System.Object ReadReferencingElement(
   System.String name,
   System.String ns,
   System.String& fixupReference
);
protected System.Object ReadReferencingElement(
   System.String name,
   System.String ns,
   boolean elementCanBeType,
   System.String& fixupReference
);
protected System.Xml.Serialization.IXmlSerializable ReadSerializable(
   System.Xml.Serialization.IXmlSerializable serializable
);
protected System.String ReadString(
   System.String value
);
protected System.Object ReadTypedPrimitive(
   System.Xml.XmlQualifiedName type
);
protected System.Xml.XmlNode ReadXmlNode(
   boolean wrapped
);
protected void Referenced(
   System.Object o
);
protected System.Array ShrinkArray(
   System.Array a,
   int length,
   System.Type elementType,
```

```
   boolean isNullable
);
protected ubyte[] ToByteArrayBase64(
   boolean isNull
);
protected static ubyte[] ToByteArrayBase64(
   System.String value
);
protected ubyte[] ToByteArrayHex(
   boolean isNull
);
protected static ubyte[] ToByteArrayHex(
   System.String value
);
protected static char ToChar(
   System.String value
);
protected static System.DateTime ToDate(
   System.String value
);
protected static System.DateTime ToDateTime(
   System.String value
);
protected static long ToEnum(
   System.String value,
   System.Collections.Hashtable h,
   System.String typeName
);
protected static System.DateTime ToTime(
   System.String value
);
protected static System.String ToXmlName(
   System.String value
);
protected static System.String ToXmlNCName(
   System.String value
);
protected static System.String ToXmlNmToken(
   System.String value
);
protected static System.String ToXmlNmTokens(
   System.String value
);
protected System.Xml.XmlQualifiedName ToXmlQualifiedName(
   System.String value
);
protected void UnknownAttribute(
   System.Object o,
   System.Xml.XmlAttribute attr
);
protected void UnknownElement(
   System.Object o,
   System.Xml.XmlElement elem
);
protected void UnknownNode(
   System.Object o
);
protected void UnreferencedObject(
   System.String id,
   System.Object o
);
```

XmlSerializationReader.CollectionFixup Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
protected class CollectionFixup
   extends System.Object
```

Public Constructors

```
public CollectionFixup(
   System.Object collection,
   System.Xml.Serialization.XmlSerializationCollectionFixupCallback callback,
   System.Object collectionItems
);
```

Public Properties

```
public System.Xml.Serialization.XmlSerializationCollectionFixupCallback get_Callback();
public System.Object get_Collection();
public System.Object get_CollectionItems();
```

XmlSerializationReader.Fixup Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
protected class Fixup
   extends System.Object
```

Public Constructors

```
public Fixup(
   System.Object o,
   System.Xml.Serialization.XmlSerializationFixupCallback callback,
   int count
);
public Fixup(
   System.Object o,
   System.Xml.Serialization.XmlSerializationFixupCallback callback,
   System.String[] ids
);
```

Public Properties

```
public System.Xml.Serialization.XmlSerializationFixupCallback get_Callback();
public System.String[] get_Ids();
public System.Object get_Source();
public void set_Source(
   System.Object value
);
```

XmlSerializationWriteCallback Delegate (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

In Visual J# you can use the delegates in the .NET Framework, but you cannot define your own. To use this delegate, pass a method with the following signature to the delegate's constructor then call the relevant add_ or remove_ method to register that method as an event handler.

```
void XmlSerializationWriteCallback(
   System.Object o
);
```

XmlSerializationWriter Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XmlSerializationWriter
   extends System.Object
```

Protected Constructors

```
protected XmlSerializationWriter();
```

Protected Properties

```
protected System.Collections.ArrayList get_Namespaces();
protected void set_Namespaces(
   System.Collections.ArrayList value
);
protected System.Xml.XmlWriter get_Writer();
protected void set_Writer(
   System.Xml.XmlWriter value
);
```

Protected Methods

```
protected void AddWriteCallback(
   System.Type type,
   System.String typeName,
   System.String typeNs,
   System.Xml.Serialization.XmlSerializationWriteCallback callback
);
protected System.Exception CreateMismatchChoiceException(
   System.String value,
   System.String elementName,
   System.String enumValue
);
protected System.Exception CreateUnknownAnyElementException(
   System.String name,
   System.String ns
);
protected System.Exception CreateUnknownTypeException(
   System.Object o
);
protected System.Exception CreateUnknownTypeException(
   System.Type type
);
protected static ubyte[] FromByteArrayBase64(
   ubyte[] value
);
protected static System.String FromByteArrayHex(
   ubyte[] value
);
protected static System.String FromChar(
   char value
);
protected static System.String FromDate(
   System.DateTime value
);
```

```
protected static System.String FromDateTime(
   System.DateTime value
);
protected static System.String FromEnum(
   long value,
   System.String[] values,
   long[] ids
);
protected static System.String FromTime(
   System.DateTime value
);
protected static System.String FromXmlName(
   System.String name
);
protected static System.String FromXmlNCName(
   System.String ncName
);
protected static System.String FromXmlNmToken(
   System.String nmToken
);
protected static System.String FromXmlNmTokens(
   System.String nmTokens
);
protected System.String FromXmlQualifiedName(
   System.Xml.XmlQualifiedName xmlQualifiedName
);
protected abstract void InitCallbacks();
protected void TopLevelElement();
protected void WriteAttribute(
   System.String localName,
   ubyte[] value
);
protected void WriteAttribute(
   System.String localName,
   System.String value
);
protected void WriteAttribute(
   System.String localName,
   System.String ns,
   ubyte[] value
);
protected void WriteAttribute(
   System.String localName,
   System.String ns,
   System.String value
);
protected void WriteAttribute(
   System.String prefix,
   System.String localName,
   System.String ns,
   System.String value
);
protected void WriteElementEncoded(
   System.Xml.XmlNode node,
   System.String name,
   System.String ns,
   boolean isNullable,
   boolean any
);
protected void WriteElementLiteral(
   System.Xml.XmlNode node,
   System.String name,
   System.String ns,
   boolean isNullable,
   boolean any
);
protected void WriteElementQualifiedName(
   System.String localName,
```

```
   System.Xml.XmlQualifiedName value
);
protected void WriteElementQualifiedName(
   System.String localName,
   System.String ns,
   System.Xml.XmlQualifiedName value
);
protected void WriteElementQualifiedName(
   System.String localName,
   System.Xml.XmlQualifiedName value,
   System.Xml.XmlQualifiedName xsiType
);
protected void WriteElementQualifiedName(
   System.String localName,
   System.String ns,
   System.Xml.XmlQualifiedName value,
   System.Xml.XmlQualifiedName xsiType
);
protected void WriteElementString(
   System.String localName,
   System.String value
);
protected void WriteElementString(
   System.String localName,
   System.String ns,
   System.String value
);
protected void WriteElementString(
   System.String localName,
   System.String value,
   System.Xml.XmlQualifiedName xsiType
);
protected void WriteElementString(
   System.String localName,
   System.String ns,
   System.String value,
   System.Xml.XmlQualifiedName xsiType
);
protected void WriteElementStringRaw(
   System.String localName,
   ubyte[] value
);
protected void WriteElementStringRaw(
   System.String localName,
   System.String value
);
protected void WriteElementStringRaw(
   System.String localName,
   ubyte[] value,
   System.Xml.XmlQualifiedName xsiType
);
protected void WriteElementStringRaw(
   System.String localName,
   System.String ns,
   ubyte[] value
);
protected void WriteElementStringRaw(
   System.String localName,
   System.String ns,
   System.String value
);
protected void WriteElementStringRaw(
   System.String localName,
   System.String value,
   System.Xml.XmlQualifiedName xsiType
);
protected void WriteElementStringRaw(
   System.String localName,
```

```
   System.String ns,
   ubyte[] value,
   System.Xml.XmlQualifiedName xsiType
);
protected void WriteElementStringRaw(
   System.String localName,
   System.String ns,
   System.String value,
   System.Xml.XmlQualifiedName xsiType
);
protected void WriteEmptyTag(
   System.String name
);
protected void WriteEmptyTag(
   System.String name,
   System.String ns
);
protected void WriteEndElement();
protected void WriteEndElement(
   System.Object o
);
protected void WriteId(
   System.Object o
);
protected void WriteNamespaceDeclarations(
   System.Xml.Serialization.XmlSerializerNamespaces xmlns
);
protected void WriteNullableQualifiedNameEncoded(
   System.String name,
   System.String ns,
   System.Xml.XmlQualifiedName value,
   System.Xml.XmlQualifiedName xsiType
);
protected void WriteNullableQualifiedNameLiteral(
   System.String name,
   System.String ns,
   System.Xml.XmlQualifiedName value
);
protected void WriteNullableStringEncoded(
   System.String name,
   System.String ns,
   System.String value,
   System.Xml.XmlQualifiedName xsiType
);
protected void WriteNullableStringEncodedRaw(
   System.String name,
   System.String ns,
   ubyte[] value,
   System.Xml.XmlQualifiedName xsiType
);
protected void WriteNullableStringEncodedRaw(
   System.String name,
   System.String ns,
   System.String value,
   System.Xml.XmlQualifiedName xsiType
);
protected void WriteNullableStringLiteral(
   System.String name,
   System.String ns,
   System.String value
);
protected void WriteNullableStringLiteralRaw(
   System.String name,
   System.String ns,
   ubyte[] value
);
protected void WriteNullableStringLiteralRaw(
   System.String name,
```

```
   System.String ns,
   System.String value
);
protected void WriteNullTagEncoded(
   System.String name
);
protected void WriteNullTagEncoded(
   System.String name,
   System.String ns
);
protected void WriteNullTagLiteral(
   System.String name
);
protected void WriteNullTagLiteral(
   System.String name,
   System.String ns
);
protected void WritePotentiallyReferencingElement(
   System.String n,
   System.String ns,
   System.Object o
);
protected void WritePotentiallyReferencingElement(
   System.String n,
   System.String ns,
   System.Object o,
   System.Type ambientType
);
protected void WritePotentiallyReferencingElement(
   System.String n,
   System.String ns,
   System.Object o,
   System.Type ambientType,
   boolean suppressReference
);
protected void WritePotentiallyReferencingElement(
   System.String n,
   System.String ns,
   System.Object o,
   System.Type ambientType,
   boolean suppressReference,
   boolean isNullable
);
protected void WriteReferencedElements();
protected void WriteReferencingElement(
   System.String n,
   System.String ns,
   System.Object o
);
protected void WriteReferencingElement(
   System.String n,
   System.String ns,
   System.Object o,
   boolean isNullable
);
protected void WriteSerializable(
   System.Xml.Serialization.IXmlSerializable serializable,
   System.String name,
   System.String ns,
   boolean isNullable
);
protected void WriteStartDocument();
protected void WriteStartElement(
   System.String name
);
protected void WriteStartElement(
   System.String name,
   System.String ns
```

```
);
protected void WriteStartElement(
   System.String name,
   System.String ns,
   boolean writePrefixed
);
protected void WriteStartElement(
   System.String name,
   System.String ns,
   System.Object o
);
protected void WriteStartElement(
   System.String name,
   System.String ns,
   System.Object o,
   boolean writePrefixed
);
protected void WriteTypedPrimitive(
   System.String name,
   System.String ns,
   System.Object o,
   boolean xsiType
);
protected void WriteValue(
   ubyte[] value
);
protected void WriteValue(
   System.String value
);
protected void WriteXmlAttribute(
   System.Xml.XmlNode node
);
protected void WriteXmlAttribute(
   System.Xml.XmlNode node,
   System.Object container
);
protected void WriteXsiType(
   System.String name,
   System.String ns
);
```

XmlSerializer Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSerializer
   extends System.Object
```

Public Constructors

```
public XmlSerializer(
   System.Type type
);
public XmlSerializer(
   System.Xml.Serialization.XmlTypeMapping xmlTypeMapping
);
public XmlSerializer(
   System.Type type,
   System.String defaultNamespace
);
public XmlSerializer(
   System.Type type,
   System.Type[] extraTypes
);
public XmlSerializer(
   System.Type type,
   System.Xml.Serialization.XmlAttributeOverrides overrides
);
public XmlSerializer(
   System.Type type,
   System.Xml.Serialization.XmlRootAttribute root
);
public XmlSerializer(
   System.Type type,
   System.Xml.Serialization.XmlAttributeOverrides overrides,
   System.Type[] extraTypes,
   System.Xml.Serialization.XmlRootAttribute root,
   System.String defaultNamespace
);
```

Public Methods

```
public boolean CanDeserialize(
   System.Xml.XmlReader xmlReader
);
public System.Object Deserialize(
   System.IO.Stream stream
);
public System.Object Deserialize(
   System.IO.TextReader textReader
);
public System.Object Deserialize(
   System.Xml.XmlReader xmlReader
);
public static System.Xml.Serialization.XmlSerializer[] FromMappings(
   System.Xml.Serialization.XmlMapping[] mappings
);
public static System.Xml.Serialization.XmlSerializer[] FromTypes(
   System.Type[] types
);
```

```
public void Serialize(
   System.IO.Stream stream,
   System.Object o
);
public void Serialize(
   System.IO.TextWriter textWriter,
   System.Object o
);
public void Serialize(
   System.Xml.XmlWriter xmlWriter,
   System.Object o
);
public void Serialize(
   System.IO.Stream stream,
   System.Object o,
   System.Xml.Serialization.XmlSerializerNamespaces namespaces
);
public void Serialize(
   System.IO.TextWriter textWriter,
   System.Object o,
   System.Xml.Serialization.XmlSerializerNamespaces namespaces
);
public void Serialize(
   System.Xml.XmlWriter xmlWriter,
   System.Object o,
   System.Xml.Serialization.XmlSerializerNamespaces namespaces
);
```

Public Events

```
public void add_UnknownAttribute(
   System.Xml.Serialization.XmlAttributeEventHandler value
);
public void remove_UnknownAttribute(
   System.Xml.Serialization.XmlAttributeEventHandler value
);
public void add_UnknownElement(
   System.Xml.Serialization.XmlElementEventHandler value
);
public void remove_UnknownElement(
   System.Xml.Serialization.XmlElementEventHandler value
);
public void add_UnknownNode(
   System.Xml.Serialization.XmlNodeEventHandler value
);
public void remove_UnknownNode(
   System.Xml.Serialization.XmlNodeEventHandler value
);
public void add_UnreferencedObject(
   System.Xml.Serialization.UnreferencedObjectEventHandler value
);
public void remove_UnreferencedObject(
   System.Xml.Serialization.UnreferencedObjectEventHandler value
);
```

Protected Constructors

```
protected XmlSerializer();
```

Protected Methods

```
protected System.Xml.Serialization.XmlSerializationReader CreateReader();
```

```
protected System.Xml.Serialization.XmlSerializationWriter CreateWriter();
protected System.Object Deserialize(
   System.Xml.Serialization.XmlSerializationReader reader
);
protected void Serialize(
   System.Object o,
   System.Xml.Serialization.XmlSerializationWriter writer
);
```

XmlSerializerNamespaces Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlSerializerNamespaces
   extends System.Object
```

Public Constructors

```
public XmlSerializerNamespaces();
public XmlSerializerNamespaces(
   System.Xml.Serialization.XmlSerializerNamespaces namespaces
);
public XmlSerializerNamespaces(
   System.Xml.XmlQualifiedName[] namespaces
);
```

Public Properties

```
public int get_Count();
```

Public Methods

```
public void Add(
   System.String prefix,
   System.String ns
);
public System.Xml.XmlQualifiedName[] ToArray();
```

XmlTextAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Property, Field, Parameter, ReturnVal
ue, AllowMultiple = false, Inherited = true) */
public class XmlTextAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlTextAttribute();
public XmlTextAttribute(
   System.Type type
);
```

Public Properties

```
public System.String get_DataType();
public void set_DataType(
   System.String value
);
public System.Type get_Type();
public void set_Type(
   System.Type value
);
```

XmlTypeAttribute Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.AttributeUsageAttribute(ValidOn = Class, Struct, Enum, Interface, Allow
Multiple = false, Inherited = true) */
public class XmlTypeAttribute
   extends System.Attribute
```

Public Constructors

```
public XmlTypeAttribute();
public XmlTypeAttribute(
   System.String typeName
);
```

Public Properties

```
public boolean get_IncludeInSchema();
public void set_IncludeInSchema(
   boolean value
);
public System.String get_Namespace();
public void set_Namespace(
   System.String value
);
public System.String get_TypeName();
public void set_TypeName(
   System.String value
);
```

XmlTypeMapping Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XmlTypeMapping
   extends System.Xml.Serialization.XmlMapping
```

Public Properties

```
public System.String get_ElementName();
public System.String get_Namespace();
public System.String get_TypeFullName();
public System.String get_TypeName();
```

System.Xml.XPath Namespace (Visual J# Syntax)

Classes

XPathDocument

XPathException

XPathExpression

XPathNavigator

XPathNodeIterator

Interfaces

IXPathNavigable

Enumerations

XmlCaseOrder

XmlDataType

XmlSortOrder

XPathNamespaceScope

XPathNodeType

XPathResultType

IXPathNavigable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IXPathNavigable
```

Public Methods

```
System.Xml.XPath.XPathNavigator CreateNavigator();
```

XmlCaseOrder Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlCaseOrder
```

Members

LowerFirst

None

UpperFirst

XmlDataType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlDataType
```

Members

Number

Text

XmlSortOrder Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XmlSortOrder
```

Members

Ascending

Descending

XPathDocument Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XPathDocument
   extends System.Object
   implements System.Xml.XPath.IXPathNavigable
```

Public Constructors

```
public XPathDocument(
   System.IO.Stream stream
);
public XPathDocument(
   System.IO.TextReader reader
);
public XPathDocument(
   System.String uri
);
public XPathDocument(
   System.Xml.XmlReader reader
);
public XPathDocument(
   System.String uri,
   System.Xml.XmlSpace space
);
public XPathDocument(
   System.Xml.XmlReader reader,
   System.Xml.XmlSpace space
);
```

Public Methods

```
public final System.Xml.XPath.XPathNavigator CreateNavigator();
```

XPathException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class XPathException
   extends System.SystemException
```

Public Constructors

```
public XPathException(
   System.String message,
   System.Exception innerException
);
```

Public Properties

```
public System.String get_Message();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected XPathException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

XPathExpression Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XPathExpression
   extends System.Object
```

Public Properties

```
public abstract System.String get_Expression();
public abstract System.Xml.XPath.XPathResultType get_ReturnType();
```

Public Methods

```
public abstract void AddSort(
   System.Object expr,
   System.Collections.IComparer comparer
);
public abstract void AddSort(
   System.Object expr,
   System.Xml.XPath.XmlSortOrder order,
   System.Xml.XPath.XmlCaseOrder caseOrder,
   System.String lang,
   System.Xml.XPath.XmlDataType dataType
);
public abstract System.Xml.XPath.XPathExpression Clone();
public abstract void SetContext(
   System.Xml.XmlNamespaceManager nsManager
);
```

XPathNamespaceScope Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XPathNamespaceScope
```

Members

All

ExcludeXml

Local

XPathNavigator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XPathNavigator
   extends System.Object
   implements System.ICloneable
```

Public Properties

```
public abstract System.String get_BaseURI();
public abstract boolean get_HasAttributes();
public abstract boolean get_HasChildren();
public abstract boolean get_IsEmptyElement();
public abstract System.String get_LocalName();
public abstract System.String get_Name();
public abstract System.String get_NamespaceURI();
public abstract System.Xml.XmlNameTable get_NameTable();
public abstract System.Xml.XPath.XPathNodeType get_NodeType();
public abstract System.String get_Prefix();
public abstract System.String get_Value();
public abstract System.String get_XmlLang();
```

Public Methods

```
public abstract System.Xml.XPath.XPathNavigator Clone();
public System.Xml.XmlNodeOrder ComparePosition(
   System.Xml.XPath.XPathNavigator nav
);
public System.Xml.XPath.XPathExpression Compile(
   System.String xpath
);
public System.Object Evaluate(
   System.String xpath
);
public System.Object Evaluate(
   System.Xml.XPath.XPathExpression expr
);
public System.Object Evaluate(
   System.Xml.XPath.XPathExpression expr,
   System.Xml.XPath.XPathNodeIterator context
);
public abstract System.String GetAttribute(
   System.String localName,
   System.String namespaceURI
);
public abstract System.String GetNamespace(
   System.String name
);
public boolean IsDescendant(
   System.Xml.XPath.XPathNavigator nav
);
public abstract boolean IsSamePosition(
   System.Xml.XPath.XPathNavigator other
);
public boolean Matches(
   System.String xpath
);
public boolean Matches(
```

```
      System.Xml.XPath.XPathExpression expr
   );
   public abstract boolean MoveTo(
      System.Xml.XPath.XPathNavigator other
   );
   public abstract boolean MoveToAttribute(
      System.String localName,
      System.String namespaceURI
   );
   public abstract boolean MoveToFirst();
   public abstract boolean MoveToFirstAttribute();
   public abstract boolean MoveToFirstChild();
   public boolean MoveToFirstNamespace();
   public abstract boolean MoveToFirstNamespace(
      System.Xml.XPath.XPathNamespaceScope namespaceScope
   );
   public abstract boolean MoveToId(
      System.String id
   );
   public abstract boolean MoveToNamespace(
      System.String name
   );
   public abstract boolean MoveToNext();
   public abstract boolean MoveToNextAttribute();
   public boolean MoveToNextNamespace();
   public abstract boolean MoveToNextNamespace(
      System.Xml.XPath.XPathNamespaceScope namespaceScope
   );
   public abstract boolean MoveToParent();
   public abstract boolean MoveToPrevious();
   public abstract void MoveToRoot();
   public System.Xml.XPath.XPathNodeIterator Select(
      System.String xpath
   );
   public System.Xml.XPath.XPathNodeIterator Select(
      System.Xml.XPath.XPathExpression expr
   );
   public System.Xml.XPath.XPathNodeIterator SelectAncestors(
      System.Xml.XPath.XPathNodeType type,
      boolean matchSelf
   );
   public System.Xml.XPath.XPathNodeIterator SelectAncestors(
      System.String name,
      System.String namespaceURI,
      boolean matchSelf
   );
   public System.Xml.XPath.XPathNodeIterator SelectChildren(
      System.Xml.XPath.XPathNodeType type
   );
   public System.Xml.XPath.XPathNodeIterator SelectChildren(
      System.String name,
      System.String namespaceURI
   );
   public System.Xml.XPath.XPathNodeIterator SelectDescendants(
      System.Xml.XPath.XPathNodeType type,
      boolean matchSelf
   );
   public System.Xml.XPath.XPathNodeIterator SelectDescendants(
      System.String name,
      System.String namespaceURI,
      boolean matchSelf
   );
   public System.String ToString();
```

Protected Constructors

```
protected XPathNavigator();
```

XPathNodeIterator Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XPathNodeIterator
   extends System.Object
   implements System.ICloneable
```

Public Properties

```
public int get_Count();
public abstract System.Xml.XPath.XPathNavigator get_Current();
public abstract int get_CurrentPosition();
```

Public Methods

```
public abstract System.Xml.XPath.XPathNodeIterator Clone();
public abstract boolean MoveNext();
```

Protected Constructors

```
protected XPathNodeIterator();
```

XPathNodeType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XPathNodeType
```

Members

All

Attribute

Comment

Element

Namespace

ProcessingInstruction

Root

SignificantWhitespace

Text

Whitespace

XPathResultType Enumeration (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public enum XPathResultType
```

Members

Any

Boolean

Error

Navigator

NodeSet

Number

String

System.Xml.Xsl Namespace (Visual J# Syntax)

Classes

XsltArgumentList

XsltCompileException

XsltContext

XsltException

XslTransform

Interfaces

IXsltContextFunction

IXsltContextVariable

IXsltContextFunction Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IXsltContextFunction
```

Public Properties

```
System.Xml.XPath.XPathResultType[] get_ArgTypes();
int get_Maxargs();
int get_Minargs();
System.Xml.XPath.XPathResultType get_ReturnType();
```

Public Methods

```
System.Object Invoke(
   System.Xml.Xsl.XsltContext xsltContext,
   System.Object[] args,
   System.Xml.XPath.XPathNavigator docContext
);
```

IXsltContextVariable Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public interface IXsltContextVariable
```

Public Properties

```
boolean get_IsLocal();
boolean get_IsParam();
System.Xml.XPath.XPathResultType get_VariableType();
```

Public Methods

```
System.Object Evaluate(
   System.Xml.Xsl.XsltContext xsltContext
);
```

XsltArgumentList Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XsltArgumentList
   extends System.Object
```

Public Constructors

```
public XsltArgumentList();
```

Public Methods

```
public void AddExtensionObject(
   System.String namespaceUri,
   System.Object extension
);
public void AddParam(
   System.String name,
   System.String namespaceUri,
   System.Object parameter
);
public void Clear();
public System.Object GetExtensionObject(
   System.String namespaceUri
);
public System.Object GetParam(
   System.String name,
   System.String namespaceUri
);
public System.Object RemoveExtensionObject(
   System.String namespaceUri
);
public System.Object RemoveParam(
   System.String name,
   System.String namespaceUri
);
```

XsltCompileException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class XsltCompileException
   extends System.Xml.Xsl.XsltException
```

Public Constructors

```
public XsltCompileException(
   System.Exception inner,
   System.String sourceUri,
   int lineNumber,
   int linePosition
);
```

Public Properties

```
public System.String get_Message();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected XsltCompileException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

XsltContext Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public abstract class XsltContext
   extends System.Xml.XmlNamespaceManager
```

Public Constructors

```
public XsltContext();
public XsltContext(
   System.Xml.NameTable table
);
```

Public Properties

```
public abstract boolean get_Whitespace();
```

Public Methods

```
public abstract int CompareDocument(
   System.String baseUri,
   System.String nextbaseUri
);
public abstract boolean PreserveWhitespace(
   System.Xml.XPath.XPathNavigator node
);
public abstract System.Xml.Xsl.IXsltContextFunction ResolveFunction(
   System.String prefix,
   System.String name,
   System.Xml.XPath.XPathResultType[] ArgTypes
);
public abstract System.Xml.Xsl.IXsltContextVariable ResolveVariable(
   System.String prefix,
   System.String name
);
```

XsltException Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
/** @attribute System.SerializableAttribute() */
public class XsltException
   extends System.SystemException
```

Public Constructors

```
public XsltException(
   System.String message,
   System.Exception innerException
);
```

Public Properties

```
public int get_LineNumber();
public int get_LinePosition();
public System.String get_Message();
public System.String get_SourceUri();
```

Public Methods

```
public void GetObjectData(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

Protected Constructors

```
protected XsltException(
   System.Runtime.Serialization.SerializationInfo info,
   System.Runtime.Serialization.StreamingContext context
);
```

XslTransform Members (Visual J# Syntax)

This topic shows Visual J# syntax for the .NET Framework library. For more information, click the link to the corresponding .NET Framework topic.

Declaration

```
public class XslTransform
   extends System.Object
```

Public Constructors

```
public XslTransform();
```

Public Properties

```
public void set_XmlResolver(
   System.Xml.XmlResolver value
);
```

Public Methods

```
public void Load(
   System.String url
);
public void Load(
   System.Xml.XPath.IXPathNavigable stylesheet
);
public void Load(
   System.Xml.XPath.XPathNavigator stylesheet
);
public void Load(
   System.Xml.XmlReader stylesheet
);
public void Load(
   System.String url,
   System.Xml.XmlResolver resolver
);
public void Load(
   System.Xml.XPath.IXPathNavigable stylesheet,
   System.Xml.XmlResolver resolver
);
public void Load(
   System.Xml.XPath.XPathNavigator stylesheet,
   System.Xml.XmlResolver resolver
);
public void Load(
   System.Xml.XmlReader stylesheet,
   System.Xml.XmlResolver resolver
);
public System.Xml.XmlReader Transform(
   System.Xml.XPath.IXPathNavigable input,
   System.Xml.Xsl.XsltArgumentList args
);
public System.Xml.XmlReader Transform(
   System.Xml.XPath.XPathNavigator input,
   System.Xml.Xsl.XsltArgumentList args
);
public void Transform(
   System.String inputfile,
```

```
      System.String outputfile
   );
   public void Transform(
      System.Xml.XPath.IXPathNavigable input,
      System.Xml.Xsl.XsltArgumentList args,
      System.IO.Stream output
   );
   public void Transform(
      System.Xml.XPath.IXPathNavigable input,
      System.Xml.Xsl.XsltArgumentList args,
      System.IO.TextWriter output
   );
   public void Transform(
      System.Xml.XPath.IXPathNavigable input,
      System.Xml.Xsl.XsltArgumentList args,
      System.Xml.XmlWriter output
   );
   public void Transform(
      System.Xml.XPath.XPathNavigator input,
      System.Xml.Xsl.XsltArgumentList args,
      System.IO.Stream output
   );
   public void Transform(
      System.Xml.XPath.XPathNavigator input,
      System.Xml.Xsl.XsltArgumentList args,
      System.IO.TextWriter output
   );
   public void Transform(
      System.Xml.XPath.XPathNavigator input,
      System.Xml.Xsl.XsltArgumentList args,
      System.Xml.XmlWriter output
   );
```

General, Visual J#, Text Editor, Options Dialog Box

The **General** property page, in the **Visual J#** folder of the **Text Editor** folder of the **Options** dialog box (**Tools** menu), contains the following properties:

Statement Completion

Auto list members
When selected, a list of available members of a given object or class appears at the appropriate places when you enter a line of code. See List Members for more information.

Hide advanced members
When selected, certain members internally marked as "advanced" are hidden from view in the statement completion list. Advanced members are those that exist only for infrastructure purposes, but must be available to view. This option allows you to filter them from view if you do not need to see them.

Parameter information
When selected, parameter information is displayed for the function or procedure, if any is available. See Parameter Info for more information.

Settings

Enable virtual space
When selected, allows you to move the cursor beyond the physical end of a line of text into character spaces that do not exist. If you move your cursor into this area and begin typing, the void between your current position and the end of the last word is automatically filled with space characters. When this setting is not selected, you cannot move the cursor beyond the last character in a line.

Word wrap
When selected, lines that extend horizontally beyond the visible range of the Text Editor are automatically wrapped to the next line.

Display

Line numbers
When selected, line numbers display to the left of each line of text.

Enable single-click URL navigation
When selected, URLs embedded in the text require only a single mouse click to go to their Web page.

Navigation bar
When selected, displays the navigation bar, which is at the top of the Code Editor. This allows you to navigate to a particular object or procedure within that object.

See Also

Text Editor, Options Dialog Box | General, Environment, Options Dialog Box

Formatting, Visual J#, Text Editor, Options Dialog Box

The **Formatting** property page, in the **Visual J#** folder of the **Text Editor** folder of the **Options** dialog box (**Tools** menu), contains the following properties:

Leave open braces on same line as construct
The Code Editor moves the open brace for a construct, such as an **if** statement, to the line after the statement declaration, aligning the open brace with the first character in the declaration. When this option is selected, the editor does not move an open brace from the declaration line.

Indent case labels
A **case** statement is aligned with the first character in a **switch** statement. When this option is selected, the Code Editor indents the **case** statements.

Automatically format completed constructs and pasted source
By default, as you type code constructs, the Code Editor can format and align them. Similarly, when you paste in code, the editor can format the text. When this option is selected, the editor formats and aligns text that you type and paste in.

Smart comment editing
When using Javadoc-style code comments, the Code Editor can complete the comment after you type the /** comment introduction. When this option is checked, the editor will insert the */ characters on the subsequent line to close the comment.

Enter outlining mode when files open
When you bring a file into the Code Editor, you can enable the outlining feature. See Outlining and Hiding Code for more information. When this option is selected, the outlining feature is enabled when you open a file.

Collapse #region blocks when files open
If one or more #region blocks are in your code, you can specify whether those blocks are open or closed when a file is open. When this option is selected, **#region** blocks are collapsed when the file is opened.

IntelliSense preselects most frequently used members
IntelliSense is designed to preselect members that you use most frequently during automatic object name completion. For more information, seeIntelliSense for Most Frequently Used Members.

When this option is selected, IntelliSense will preselect the member that you use most frequently in the pop-up List Members box.

Clear history
Clears the list of members that you have used during automatic object name completion.

See Also

Text Editor, Options Dialog Box | General, Environment, Options Dialog Box

General, Common Properties, <Projectname> Property Pages Dialog Box

The **General** property page in the **Common Properties** folder contains the following properties:

Application Group

Assembly Name

The name of the output file that will hold the assembly manifest. Changing this property will also change the **Output File** property.

See AssemblyName Property for information on how to programmatically access this property.

Output Type

The type of application to build. In a Web Application project, this property can only be set to Class Library.

See OutputType Property for information on how to programmatically access this property.

Default Package

The package for added items, such as classes, that are added via the Add New Item Dialog Box.

Startup Object

The name of the class that contains the main method that you want called on program startup. If **Not Set** is selected, the first file found that contains a main method will be used.

See /main and StartupObject Property for information on how to programmatically access this property.

Supported Runtimes

Specifies the version of the common language runtime supported by the application. Unless otherwise specified, the application will run on the version of the runtime that is used to build the application. Modify this property if you want your application to run on any version of the common language runtime.

This property is not available programmatically. It is only available for Windows Application and ASP.NET Web Application projects. For a Windows application, selecting a runtime causes the project system to update the app.config file in your project or to create an app.config file if one does not already exist. At runtime, the name of the app.config file must be *filename.ext*.config (where *filename.ext* is the name of the executable that started the application) and the file must be in the same directory as the executable. The project system will create the *filename.ext*.config file from the app.config file and place it in the bin*target* directory.

If you specify more than one runtime version and the Windows application runs on a computer with more than one runtime version installed, the first version specified in the .config file that matches an installed runtime that is available on the system will be used.

For more information, see Targeting a .NET Framework Version.

For ASP.NET Web Application projects, modifying this property causes the project system to update the web.config file in your project by adding assembly binding (<bindingRedirect> and <assemblyIdentity>) tags in a <dependentAssembly> section. These binding statements ensure that a consistent set of .NET Framework assemblies is used for the application when it is running under an older version of the runtime. The web.config file remains in the project directory and is not renamed.

When deploying the application to an environment with more than one available version of the runtime, ensure the application's virtual directory in IIS is scriptmapped to the intended version of ASP.NET. An application can be scriptmapped to an ASP.NET version using the aspnet_regiis.exe command-line tool associated with the applicable version. See the aspnet_regiis.exe documentation for more information.

Project Group

Project File

The name of the project file for the current project. This is a read-only property while the project is open in Visual Studio.

See FileName Property for information on how to programmatically access this property.

Project Folder

The path to the project directory. While the project is open in Visual Studio, this property is read only.

See LocalPath Property and FullPath Property for information on how to programmatically access this property.

Output File

The name of the project's primary output file (also known as the program image). The file that holds the assembly metadata must be the output file of this application. Therefore, **Output File** is a read-only property that can only be changed by changing the **Assembly Name** property.

See OutputFileName Property for information on how to programmatically access this property.

Wrapper Assemblies for ActiveX/COM Objects Group

Wrapper Assembly Key File

The key file to sign COM components referenced by this project. COM components can be referenced via the Add Reference dialog box. Use sn.exe to generate a key pair and then register the key pair in a container.

See AssemblyOriginatorKeyFile Property and AssemblyOriginatorKeyMode Property for information on how to programmatically access this property.

Wrapper Assembly Key Name

The key container to sign COM components referenced by this project. COM components can be referenced via the Add Reference dialog box. Use sn.exe to generate a key container.

The container does not have to be stored on your local machine; it can be on an external device connected to your computer.

See AssemblyKeyContainerName Property and AssemblyOriginatorKeyMode Property for information on how to programmatically access this property.

For information on how to access the **General** property page in the **Common Properties** folder, see Setting Visual J# Project Properties.

See Also

Visual J# Project Properties | Setting Visual J# Project Properties

Designer Defaults, Common Properties, <Projectname> Property Pages Dialog Box

The **Designer Defaults** property page in the **Common Properties** folder, which is used to specify default layout and configuration settings for your project's Web Forms designers, contains the following properties:

Web Designers Group

Page Layout

Specifies whether Web Forms pages added to the project should be laid out in a linear (one-dimensional) or grid (two-dimensional) format by default. This project property can be overridden on a file by means of the Properties window.

See DefaultHTMLPageLayout Property for information on how to programmatically access this property.

Target Schema

Specifies the HTML platform you want to ensure that your application is compatible with.

This project property can be overridden on a file by means of the Properties window. To access this property programmatically, see DefaultTargetSchema Property. For more information on target schema, see Setting the targetSchema Property of an HTML Document.

Web Scripting Group

Client Script Language

Specifies the script language that will be used when code is generated for you to run on the client. This project property can be overridden on a file via the Properties Window.

See DefaultClientScript Property for information on how to programmatically access this property.

For information on how to access the **Designer Defaults** property page in the **Common Properties** folder, see Setting Visual J# Project Properties.

See Also

Visual J# Project Properties | Setting Visual J# Project Properties

Build, Configuration Properties, <Projectname> Property Pages Dialog Box

The **Build** property page in the **Configuration Properties** folder contains the following properties:

Code Generation Group

Conditional Compilation Constants
Specifies symbols on which to perform conditional compilation. Separate symbols with a space. See /define for more information.

See DefineConstants Property for information on how to programmatically access this property.

Optimize code
Enable or disable optimizations performed by the compiler to make your output file smaller, faster, and more efficient.

See Optimize Property for information on how to programmatically access this property.

Errors and Warnings Group

Warning Level
Specifies the level to display for compiler warnings. See /warn for more information.

See WarningLevel Property for information on how to programmatically access this property.

Treat Warnings As Errors
Treat all warnings as errors. See /warnaserror for more information.

See TreatWarningsAsErrors Property for information on how to programmatically access this property.

Suppress Specific Warnings
Block the compiler's ability to generate one or more warnings. Separate multiple warning numbers with a comma. See /nowarn for more information.

See WarningLevel Property for information on how to programmatically access this property.

Outputs Group

Output Path
Specifies the location of the output files for this project's configuration.

See OutputPath Property for information on how to programmatically access this property.

Generate Debugging Information
Generate debugging information and place it in the output file. See /debug for more information.

See DebugSymbols Property for information on how to programmatically access this property.

Register for COM interop
Indicates that your managed application will expose a COM object (a COM-callable wrapper) that allows a COM object to interact with your managed application. The **Output Type** property of this application must be **Class Library** in order for the **Register for COM interop** property to be available. For more information, see General, Common Properties, <Projectname> Property Pages Dialog Box.

For information on how to programmatically access the **Register for COM interop** property, see RegisterForComInterop Property.

For information on how to access the **Build** property page in the **Configuration Properties** folder, see Setting Visual J# Project Properties.

See Also

Visual J# Project Properties | Setting Visual J# Project Properties

Debugging, Configuration Properties, <Projectname> Property Pages Dialog Box

The **Debugging** property page in the **Configuration Properties** folder contains the following properties:

Debuggers Group

Enable ASP Debugging

Enables debugging of Active Server Pages (ASP).

See EnableASPDebugging Property for information on how to programmatically access this property.

Enable ASP.NET Debugging

Enables debugging of applications written for the ASP.NET Web development platform.

The debug attribute of the <compilation> tag in the project's web.config file specifies whether debug symbols (.pdb files) will be generated for dynamically compiled ASP.NET files such as .aspx, .asmx, and .ascx.

Because many parts of an ASP.NET application are dynamically compiled at run time, for example, .aspx and .ascx files, you must configure the ASP.NET runtime to compile the application with debugging information before these files can be debugged.

To insert debugging information into the compiled page, set debug=true.

Setting `debug=true` creates a larger file. Therefore, this setting should only be enabled when you are debugging an application, since it can significantly affect the performance of the application.

After changing this property, select the **Apply** button in order to enable the appropriate property (**Start Application**, **Start URL** or **Start Page**) for editing.

See EnableASPXDebugging Property for information on how to programmatically access this property.

Enable Unmanaged Debugging

Enables debugging of native (unmanaged) Win32 code called from your Visual J# application.

See EnableUnmanagedDebugging Property for information on how to programmatically access this property.

Enable SQL Server Debugging

Enables debugging of SQL procedures from your Visual J# application.

See EnableSQLServerDebugging Property for information on how to programmatically access this property.

Start Action Group

Debug Mode

Specifies whether you want to debug a project, a program, or a URL. You can also select **Wait to Attach to an External Process**.

- **Project** Indicates that the executable (for Windows Application and Console Application projects) or the page (for Web projects) should be started when the application is debugged. Class Library projects cannot be started directly. If you want to debug an XML Web service, setting this property will cause the application to automatically attach to Internet Explorer.
- **Program** Indicates that a specific program should be started when the application is debugged. For example, you might run a prebuilt client that consumes your Class Library project. You must set **Start Application** to debug.
- **URL** Indicates that a particular URL (**Start URL**) should be accessed when the application is debugged. For example, you might access the URL to a Web site that uses your Class Library project.
- **Wait to Attach to an External Process** This value only applies to Web Application and XML Web service projects. For an XML Web service project, the debugger will attach to a process that will call into your XML Web service. Set a breakpoint in your XML Web service before you start so that the breakpoint is hit when the external process calls into your XML Web service.

See StartAction Property for information on how to programmatically access this property.

Start Application

Specifies the command for launching the program you are debugging.

See StartProgram Property for information on how to programmatically access this property.

Start URL

If **Debug Mode** is set to URL, specifies the URL where the project you are debugging is located.

See StartURL Property for information on how to programmatically access this property.

Start Page

Specifies the URL of a page to be used as the startup page when debugging.

See StartPage Property for information on how to programmatically access this property.

Start Options Group

Command Line Arguments

Specifies arguments for the **Start Application** property.

See CommandLineArguments Property for information on how to programmatically access this property.

Working Directory

Specifies the working directory of the program being debugged.

See StartWorkingDirectory Property for information on how to programmatically access this property.

Always use Internet Explorer

Determines whether debugging will use the default Internet browser specified for Visual Studio.

See StartWithIE Property for information on how to programmatically access this property.

Enable remote debugging

When enabled, allows remote debugging of an .exe on the machine specified in the **Remote debug machine** property.

To access this property programmatically, see RemoteDebugEnabled.

Remote debug machine

When **Enable remote debugging** property is enabled, the name of the machine on which the .exe will run. The location of the .exe on the remote machine must match the value of the **Output Path** property, which is on the **Build** property page in the **Configuration Properties** folder. The remote machine needs the debugger components installed. The user of the local machine needs to be in the Debugger Users group on the remote machine. The .exe must be in a shareable directory on the remote machine.

To access this property programmatically, see RemoteDebugMachine.

For information on how to access the **Debugging** property page in the **Configuration Properties** folder, see Setting Visual J# Project Properties.

See Also

Visual J# Project Properties | Setting Visual J# Project Properties

Advanced, Configuration Properties, <Projectname> Property Pages Dialog Box

The **Advanced** property page in the **Configuration Properties** folder contains the following properties:

Base Address

Specifies the preferred base address at which to load a DLL. See /baseaddress for more information.

See BaseAddress Property for information on how to programmatically access this property.

Additional Options

You can manually specify compiler options in this box. If a compiler option is not available anywhere else in the property pages (/ss, for example), you can specify it here. For information on available compiler options, see Visual J# Compiler Options.

It is recommended that **Additional Options** be used for specifying only compiler options that are not available elsewhere in the property pages. This will prevent confusion if a value is inadvertently specified elsewhere in the property pages and in **Additional Options**. A value specified in **Additional Options** will override a value for the same property set elsewhere in the property pages. For example, if you specify Application1 as the **AssemblyName** property in the **Common Properties** folder of the **General** property page (General, Common Properties), and also specify **/out:MyProgram.exe** in the **Additional Options** box, the output assembly created by the project will be MyProgram.exe.

For information on how to access the **Advanced** property page in the **Configuration Properties** folder, see Setting Visual J# Project Properties.

See Also

Visual J# Project Properties | Setting Visual J# Project Properties

Application Samples

The samples in this section are modeled after complete applications that you might develop. These samples illustrate more complete error handling and object-oriented development than other Visual J# samples.

Sample	Description
Acronym Web Service	Demonstrates creating and consuming an XML Web service from a Web application. Also shows how to bind the contents of a dataset to a server-side Datagrid.
Agent Explorer	Demonstrates how to upgrade a Java-language/COM application to run on Visual J#.
Dice	Illustrates how to consume C# classes from Visual J#.
Photo Album	A Windows Forms photo-album application that you can use to create, open, and modify JPEG, GIF, and BMP images.
TypeFinder	Provides a command line interface to get information about types from user's environment. This sample also provides general information about reflection.
Vjsresgen	Converts a Visual J++ 6.0 resource file to a .NET Framework resource file (.resx).
WinTalk	Offers a brief introduction to Windows Forms and Sockets including other FCL topics. The sample is a very simple Sockets Chat application.
WordCount	Offers an introduction to the .NET Framework class library. Provides an application that counts words in a text file and organizes the results according to command-line arguments.

See Also

Visual J# Samples | Technology Samples | General Samples | Cross-Language Samples

Dice Sample (Consume C# Classes)

This sample illustrates how to consume C# classes from Visual J# by means of the following:

- Implementing an interface (defined in C#) on a Visual J# class.
- Handling exceptions thrown from a C# class, in Visual J#.
- Examining properties exposed by a C# class, in Visual J#.
- Sinking events raised by a C# class in Visual J#.

The cscServer project implements a die (the kind used in games).

The number of sides (faces) on the die can be set programmatically. The die is rolled by the **Roll** method. The **Roll** operation returns the number on the top face of the die.

The die uses the interface **IRoll** to perform the roll operation. Clients are required to provide an implementation of this interface. The client is required to set this interface on the die before invoking a **Roll** method on the die.

If the die is constructed with an invalid number of sides (for example, the number of sides being less than or equal to 0), the die throws a **BadDieSizeException**. If the **Roll** method on the die is invoked prior to the number of sides being set, the die throws a **System.NullReference** exception.

The die exposes a read-only property (**NumSides**) that can be used to inspect the number of sides for the die.

Every time an implementation of the **IRoll** interface is set on the die, it raises the **LoadedDice** event.

The Dice project does the following:

- Provides implementations for the **IRoll** interface.
- Catches both the exceptions thrown.
- Examines the **NumSides** property.
- Provides a delegate object to sink the events raised by the die.

Building and Running the Sample

To build from the development environment

1. Open the Dice.sln file.
2. Build the sample (CTRL+SHIFT+B).

To build from the command line

- Type **BUILD.bat**.

To run this sample

- From the development environment, press F5.

  -or-

- From the command line, type **Dice** and press ENTER.

Command-Line Requirements

For the Visual J# samples to function properly when building and running from the command line, you need to run vsvars.bat (located in the <%Program Files%>\Microsoft Visual Studio .NET 2003\Common7\Tools directory). Alternatively, you can also use the Visual Studio .NET command prompt (click **Start**, then **Programs**, then **Microsoft Visual Studio .NET 2003**, and finally **Visual Studio .NET Tools**).

For the Visual J# samples to function properly when building and running from the command line, you must additionally set the **Path** variable to include the <*Microsoft Visual J++ 6.0 install root*> directory.

See Also

Application Samples

Winforms-Photo-Album Sample (Use Windows Forms Library)

This sample demonstrates the use of the Windows Forms library using Visual J#. The sample is a photo album application that you can use to create, open, and modify new and existing album files. It opens images of type JPEG, GIF, and BMP.

Building and Running the Sample

To build and run the sample

1. Open winforms_photo_album.sln in the development environment.
2. From the **Build** menu, click **Build**.
3. From the **Debug** menu, click **Start Without Debugging**.

To create a new album

1. From the **File** menu, click **New Album**.
2. Click the **Browse** button and add the files of your choice.
3. Once the file names are displayed in the text box adjacent to the **Browse** button, click the **Add** button to add the files to the album that is being created.
4. Optional. Modify the title and/or the description for an image as follows:
   a. Select the image.
   b. Make the necessary changes in the title and/or description.
   c. Click the **Modify** button.
5. (Optional.) Remove files that are not needed by clicking the **Remove** button.
6. After you are done with editing, click the **Finish** button to create the album.
7. Enter the name of the album in the dialog box that is displayed up and save it.

   The album is ready for browsing.

Viewing the Album

There are three ways to view the album.

Auto Slide Show

- Click **Slide Show** from the **View** menu.
- The **First** and **Last** buttons take you to the first and last images in the album, respectively.
- The **Previous** and **Next** buttons take you one step backward or forward, except at the first and last images.
- The box at the top of the image displays the title, and the box at the right displays the description of the image currently being viewed.

Thumbnail View

- Click **Thumbnails** from the **View** menu.
- The images are displayed as thumbnails in the upcoming pane. Clicking them once displays a larger image at the left. Double-clicking opens a new window that displays the image in a larger size.
- To display only the name of the file as a list without any images, click **Settings** on the **View** menu and then click **List**. A similar action is performed if you click **Settings** on the **View** menu and then click **Small icons**, as there are no small icons available.

Manually

- Click **Auto Slide** on the **View** menu.
- Click the **Start** button to start the slide show.
- Click the **Pause** button to freeze the display or the **Stop** button to stop the slide show.
- You can set the speed at which images display using the **Trackbar** control.

Classes in the Sample

The following classes are used in this sample.

File	Description
Form1.jsl	This is the main class. It is a subclass of **System.Windows.Forms.Form**. It contains most of the GUI controls needed for the application.
CreateAlbum.jsl	This class contains the GUI and the functionality for creating a new album.
PicBox.jsl	A customized picture box class needed for the application.
SliderThread.jsl	This class implements the threading functionality that controls the speed at which images are displayed in an Auto Slide Show.
MesgBox.jsl	A utility class used for displaying customized error/status messages.

Command-Line Requirements

For the Visual J# samples to function properly when building and running from the command line, you need to run vsvars.bat (located in the <%Program Files%>\Microsoft Visual Studio .NET 2003\Common7\Tools directory). Alternatively, you can also use the Visual Studio .NET command prompt (click **Start**, then **Programs**, then **Microsoft Visual Studio .NET 2003**, and finally **Visual Studio .NET Tools**).

For the Visual J# samples to function properly when building and running from the command line, you must additionally set the **Path** variable to include the <*Microsoft Visual J++ 6.0 install root*> directory.

See Also

Application Samples

TypeFinder Sample (Get Type Information)

This sample provides a command-line interface to get information about types in the environment. Sometimes locating a specific type's namespace or DLL can be difficult, and once you locate the type, the documentation may be unavailable or out of date. This utility provides a simple way to determine what types are available, what module they are in, and what interfaces, methods, fields, properties, and events are available on the type. This sample also provides an introduction to reflection.

Building and Running the Sample

To build and run the sample from the development environment

1. Open the TypeFinder.sln file.
2. Build the sample (CTRL+SHIFT+B).
3. Press F5 to run the sample.

To build and run the sample from the command line

1. Type **BUILD.bat**.
2. Type **TypeFinder** and press ENTER.
3. To find the location of each type with a word in its name, type **TypeFinder** *word* from the command line.

Classes and Technologies in the Sample

The following classes and technologies are used in this sample.

Reflection

Class/Technology	Description
System.Reflection.Assembly	Loads assemblies into the AppDomain so that they can be searched for typ es.
System.Reflection.Module	Gets types from the assembly or module for comparison to the search stri ng.
System.Type	Gets type information such as name, namespace, and members.
System.Reflection.PropertyInfo	Finds information about properties in types.
System.Reflection.EventInfo	Finds information about events in types.
System.Reflection.FieldInfo	Finds information about fields in types.
System.Reflection.MethodInfo	Finds information about methods in types.

I/O

Class/Technology	Description
System.IO.TextWriter	Used by the `IndentedWriter` sample type to output (to the console, by def ault) in a generic fashion.

Text

Class/Technology	Description
System.Text.StringBuilder	Used by the `IndentedWriter` sample type to create a string.
System.String	Used throughout the sample to find format strings, find substrings, upperc ase strings, for example.

Collections

Class/Technology	Description
System.Collections.ArrayList	Used by the sample to manage a list of strings .

Registry

Class/Technology	Description
Microsoft.Win32.Registry	Used to create an instance of the `RegistryKey` type for a subkey of the `Loc alMachine` key.
Microsoft.Win32.RegistryKey	Used to read values from a key in the registry.

Command-Line Requirements

For the Visual J# samples to function properly when building and running from the command line, you need to run vsvars.bat (located in the <%Program Files%>\Microsoft Visual Studio .NET 2003\Common7\Tools directory). Alternatively, you can also use the Visual Studio .NET command prompt (click **Start**, then **Programs**, then **Microsoft Visual Studio .NET 2003**, and finally **Visual Studio .NET Tools**).

For the Visual J# samples to function properly when building and running from the command line, you must additionally set the **Path** variable to include the <*Microsoft Visual J++ 6.0 install root*> directory.

See Also

Application Samples

WinTalk Sample (Introduces Windows Forms and Sockets)

This sample offers a brief introduction to Windows Forms and Sockets including other Framework Class Library (FCL) topics. The sample is a very simple Sockets Chat application.

Building and Running the Sample

To build this sample

- Type **build.bat** from the command line.

To run this sample

- Type **WinTalk.exe /?** in the command line for a complete list of command line options for the utility.
- To try the sample where the client and the server are on the same machine, simply run the application twice with no parameters.
- When specifying a port number, verify that port is not already in use by another program; otherwise, the sample will not run correctly. Try using a high port number between 5001 and 5150.
- For the sample to function properly, your **Path**, **Include**, and **Lib** environment variables must be set correctly. You can set these variables by running CorVars.bat, located in the <*SDKRoot*>*Bin* directory. CorVars.bat must be executed in every command shell.

Classes and Technologies in the Sample

The following classes and technologies are used in this sample.

Windows Forms

Class/Technology	Description
Application	Used by the sample application to implement a message pump.
Splitter	Control type that handles splitter semantics. Used in the application to separate the send and receive text frames.
Panel	Allows for flexible organizing, docking, and anchoring of controls on a form. In this case, the panel contains the two text boxes and the splitter. The status control, on the other hand, is a p eer to the panel.
TextBox	Implements two edit controls in the sample. One is read-only for receiving text.
Label	Used to display static status information in the sample.
MessageBox	Used by the sample to relay information about errors and shutdown situations.

Networking

Class/Technology	Description
IPEndPoint	Used by the sample to encapsulate an IP address and port number.
IPAddress	Provides useful **IPAddress**-related logic.
Dns	Used to get an IP address from a DNS name.
Socket	Most networking implemented in managed code is done using an abstraction such as XML Web services. However, it is possible to implement socket code directly, and this sample uses the **Socket** class to do so.
NetworkStream	Derived from **stream**, and uses a socket as its underlying device. This makes it very useful fo r streaming data over a socket. You can use the **NetworkStream** type just like you would us e any other stream type, such as **FileStream**.

I/O

Class/Technology	Description
StreamWriter	Used to write to the network stream that represents a socket.
StreamReader	Used to read from the network stream that represents a socket .

Garbage Collection

Class/Technology	Description

GC	Used to suppress finalization of a custom object that has already been disposed .

Delegates

Class/Technology	Description
MulticastDelegate	This sample implements a delegate type derived from **MulticastDelegate**. This type is used for notifications related to network events. The sample uses this delegate type to bind the net working type to the form type in the sample

Threading

Class/Technology	Description
ThreadPool	The custom *Talker* class in this sample handles networking logic and uses a second thread fo r reading from the network. However, rather than creating a thread, it simply uses a thread fr om the common language runtime thread pool. This is the suggested method for multithrea ding in managed code.

Exceptions

Class/Technology	Description
IOException	The **StreamReader** and **StreamWriter** types commonly throw the **IOException** type. Howe ver, to catch and handle these exceptions properly, it is commonly necessary to check the att ached exception that reflects the more base error. In this sample's case this will be the **Socke tException** type.
SocketException	This sample catches the **SocketException** type and checks the **ErrorCode** property of the s ocket for certain expected error types.

Command-Line Requirements

For the Visual J# samples to function properly when building and running from the command line, your **Path** environment variable must be set correctly. You must set the **Path** variable to include the <%windir%>\Microsoft .NET\Framework\v<%version%> directory. Additionally, you may also need to run CorVars.bat (located in the <*.NETFrameworkSDKRoot*>*Bin* directory).

See Also

Application Samples

WordCount Sample (Introduction to .NET Framework Class Library)

This sample offers an introduction to the .NET Framework class library. It shows how to create an application that opens multiple files (specified on the command line) and counts each file's size (in bytes), characters, words, and lines. The results from each file and the total of all files are displayed.

Building and Running the Sample

To build and run this sample from the development environment

1. Open the WordCount.sln file.
2. Build the sample (CTRL+SHIFT+B).
3. Press F5 to run the sample.

To build and run this sample from the command line

1. Type **BUILD.bat**.
2. Type **WordCount** and press ENTER.
3. To count the words in a file, type **WordCount** *filename*.

Classes and Technologies in the Sample

The following classes and technologies are used in this sample.

I/O

Class/Technology	Description
System.IO.FileStream	Used for file access. This class is used to read to and write from files.
System.IO.StreamWriter	Used when the command-line arguments request that the output be sent to a file. The **StreamWriter** is used in conjunction with a **FileStream** instance to format text output to a file.
System.IO.StreamReader	Used to read the contents of a file for which the sample is counting words. The **StreamReader** is used in conjunction with a **FileStream** instance to read text from a file.

Collections

Class/Technology	Description
System.Collections.ArrayList	Used as a general collection class for storing sets of objects.
System.Collections.SortedList	Used as a collection class for storing sets of objects in a sorted manner.
System.Collections.IEnumerator	Used directly to enumerate over sets of objects.
System.Collections.IDictionaryEnumerator	Used directly to enumerate over indexed sets of objects.

Command-Line Requirements

For the Visual J# samples to function properly when building and running from the command line, you need to run vsvars.bat (located in the <%Program Files%>\Microsoft Visual Studio .NET 2003\Common7\Tools directory). Alternatively, you can also use the Visual Studio .NET command prompt (click **Start**, then **Programs**, then **Microsoft Visual Studio .NET 2003**, and finally **Visual Studio .NET Tools**).

For the Visual J# samples to function properly when building and running from the command line, you must additionally set the **Path** variable to include the <*Microsoft Visual J++ 6.0 install root*> directory.

See Also

Application Samples

Cross-Language Samples

The samples in this section are ported from samples in other languages.

Sample	Description
DataSetConsumer	Demonstrates how to consume multiple-table datasets in a SPROXY-generated SOAP client. The Visual J# code creates a multiple-table dataset and exposes it through an XML Web services method.
Delegates	Illustrates use of the delegates in the Managed Extensions for C++ component. A delegate is created over a method defined in a DLL created with Visual J#.
JrnlPost	Creates a single DLL containing native business logic, an interface and type library for COM clients, and an assembly with corresponding metadata for .NET Framework clients. The Visual J# code instantiates a managed object from the DLL and calls a couple of methods on it. It illustrates that the DLL contains both COM and .NET Framework classes.
ManagedEvents	Demonstrates event handling interoperability between Managed Extensions for C++ and Visual J# objects. The Visual J# code implements both an event source and an event receiver.
MCppWrapper	Illustrates how a Managed Extensions **__gc** class can be used as a proxy class (or wrapper) for an unmanaged C++ class in a DLL. The Visual J# code is a client that reads a string and position from the console and displays a suffix.
OnlineAddressBook	Stores an address book in an XML Web service created with ATL Server. The sample has a Visual J# client that calls into the XML Web service.
TilePuzzle	Demonstrates interoperability between managed components (written with Managed Extensions for C++ and Visual J#) and native components (written with C++ using COM attributes).

See Also

Visual J# Samples | Application Samples | Technology Samples | General Samples

Delegates Sample (Custom Delegates)

This sample implements two uses of a delegate (single-cast and multicast) using the **__delegate** keyword. For more information on delegates, see Delegates in Managed Extensions for C++.

The sample also demonstrates interoperability between a Managed Extensions for C++ application and a Visual J# DLL (`jsdel`). The Visual J# DLL implements a basic Visual J# class (`jsdel`) with a custom method.

Building and Running the Sample

To build and run this sample using Visual Studio

1. In the Visual Studio IDE, load the solution file Delegates.sln.
2. In **Solution Explorer**, right-click the **Delegates** solution.
3. On the shortcut menu, click **Build Solution**.
4. On the **Debug** menu, click **Start**.

When the sample is built and run, single-cast and multicast delegates are created and invoked:

- `pSCDelegate`

  A single-cast delegate is created and then the bound method is invoked. The confirmation is a printed string with the value passed by the single-cast delegate.

- `pMCDelegate`

  Before creating the multicast delegate, the Visual J# class (`jsdel`) is first instantiated. The delegate is then bound to both the Visual J# and managed class methods and invoked. The confirmation is a printed string with the value passed by the multicast delegate.

See Also

Cross-Language Samples

ManagedEvents Sample (Demonstrates Creating and Using Events in a Managed Application)

This sample demonstrates event handling interoperability between Managed Extensions for C++ and Visual J# objects.

Building and Running the Sample

1. Open the solution file **ManagedEvents.sln**.
2. Right-click the **Receiver** node in **Solution Explorer** and select **Set as Startup Project**.
3. From the **Build** menu, click **Build Solution**.
4. From the **Debug** menu, click **Start Without Debugging**.
5. The sample can also be run with JSEvent as startup project. Set JSEvent as the startup project.
6. Right-click the JSEvent project and select **Properties**. On the **General** tab, choose the **Console Application** output type.
7. Build and run the sample.

How the Sample Works

The code creates two event sources: A Managed Extensions for C++ event source `Event` (see Event.cpp) and a Visual J# event source `JSEvent` (see JSEvent.jsl). It also creates two event receivers: A Managed Extensions for C++ event receiver `Receiver` (see Receiver.cpp), and a Visual J# event receiver `JR2` (see JSEvent.jsl).

The Managed Extensions for C++ event source `Event` declares two events: `MyEvent` and `MyEvent2` The Visual J# event source `JEvent` declares one event: `JMyEvent`.

The Managed Extensions for C++ event receiver `Receiver` hooks:

- `Event::MyEvent` to the event handler methods `Handler1` and `Handler2`
- `Event::MyEvent2` to the event handler method `Handler5`
- `JEvent::JMyEvent` to the event handler methods `Handler3` and `Handler4`

The Visual J# event receiver JR2 hooks:

- `JEvent::JMyEvent` to the event handler methods `H1` and `H2`
- `Event::MyEvent` to the event handler methods `H3` and `H4`

The main code in Receiver.cpp instantiates the event sources and the Managed C++ for Extensions event receiver. Then, it calls methods to hook the event handlers to the event sources and also to fire these events.

The main code in JSEvent.jsl instantiates the event sources and the Visual J# event receiver. Then it calls methods to hook, fire, and unhook the event handlers.

The Visual J# event receiver `JR2` and the Managed Extensions for C++ event receiver `Receiver` receive events from both Managed Extensions for C++ and Visual J# event sources, `Event` and `JEvent`, respectively.

See Also

Cross-Language Samples

OnlineAddressBook Sample (Stores an Address Book in an XML Web Service)

This sample demonstrates the use of session state to store an address book in an XML Web service created with ATL Server.

The OnlineAddressBook sample uses OLE DB client template classes to store a user's address book and lets the user set and retrieve this information using an XML Web service created with ATL Server.

Requirements

This sample requires IIS and Microsoft Office XP.

Building and Running the Sample

To build and run this sample

1. Open the solution file, OnlineAddressBook.sln, in the Visual Studio development environment.
2. Modify the OnlineAddressBookWS.disco file and the deployment project target computer name to point to the computer you are planning to deploy to.

   -or-

   Run all of the projects without any modifications by deploying and running everything to localhost.
3. Copy the included Microsoft Access database, OnlineAddressBookWS.mdb, to your C:\ drive.

   -or-

   Copy it to a different location, as long as you modify `MYDATASOURCE`, which is defined in the file OnlineAddressBookWS\OnlineAddressBookWS.h.
4. Build **OnlineAddressBookWSIsapi** and **OnlineAddressBookWS**.

   A virtual directory is created on your computer, to which the relevant files are deployed.
5. Build the included Visual J# project, OnlineAddressBookJSharpClient. You can use this client to import your current personal address book from Microsoft Outlook XP and upload it to your server or any server running this XML Web service. You can also create your own contacts and upload them to the XML Web service and later download them from anywhere.
6. Build OnlineAddressBookSRF. This deploys an ATL Server project with a built-in XML Web service client. Once you have created an account and added some records to your address book with it using the Visual J# client, you can use this project to view your contacts using a browser. This page should be in http://localhost/OnlineAddressBookWS/OnlineAddressBookSRF.srf.

See Also

Cross-Language Samples

TilePuzzle Sample: Visual J# and Managed Extensions for C++ Interoperability

This sample demonstrates interoperability between managed components (written with Managed Extensions for C++ and Visual J#) and native components (written with C++ using COM attributes).

The sample implements a basic puzzle game called the Tile Puzzle. The sample loads a bitmap, splits the bitmap into any number of tiles (determined by the user), and randomizes the individual tiles. The user then solves the puzzle by sliding individual tiles around until the original picture is displayed. In addition to these features, the sample has the ability to solve the puzzle using heuristic search algorithms written in Managed Extensions for C++ and the .NET Framework classes.

Building and Running the Sample

To build and run TilePuzzle using Visual Studio

1. In the Visual Studio IDE, load the solution file Puzzle.sln.
2. In **Solution Explorer**, right-click the Puzzle solution.
3. On the shortcut menu, click **Build Solution**.
4. On the **Debug** menu, click **Start**. Verify that the Puzzle project is set as the startup project.

Remarks

After the project builds successfully, try to solve the puzzle yourself.

The save and load game features are not implemented.

To access the native COM component from within the managed .NET Framework objects, the sample uses TLBIMP.EXE to generate a .NET Framework proxy dll.

Classes in the Sample

The following classes are used in this sample.

Class	Description
System.Windows.Forms.Form	Implements the `AboutForm` object found in the Puzzle project.
System.Object	Implements the `GameLevelEnum` object found in the Puzzle project.
System.Delegate	Implements the `SolveThreadProcDlg` object found in the Puzzle project .

See Also

Cross-Language Samples

General Samples

Samples in this category include:

Sample	Description
AutoWord	Demonstrates how to automate Word from Visual J#.
CarsSelector	Demonstrates how to create a Web Forms page for selecting cars that simulates a simple e-commerce page. The sample demonstrates using the ASP.NET technology.
LocalizedHelloWorld	Demonstrates how to create and use satellite assemblies for console applications.
MultilingualForm	Demonstrates how to create and localize Windows Forms.
PerfMon	Demonstrates how to monitor system performance using performance counters and is an introduction to performance counters.
ProcessController	Demonstrates how to build a Visual J# Windows application for processes and services monitoring.
Scribble	Demonstrates a Windows Forms MDI drawing application.

See Also

Visual J# Samples | Application Samples | Technology Samples | Cross-Language Samples

AutoWord Sample (Automating Microsoft Word)

This sample includes four examples that show how to automate Microsoft Word from Visual J#. The examples demonstrate how to automate the following activities:

- Starting Word.
- Creating a new document.
- Opening an existing document.
- Hooking events in the Word application.

The sample also shows how to create an interop assembly for Microsoft Word, and shows some rudimentary techniques for automating Word.

Projects in the Sample

This sample contains five projects and produces three assemblies.

Building and Running the Sample

To create the assemblies

1. Open the solution AutoWord.sln.
2. Right-click the RunCreateWordAssembly project, then click **Build.**
3. For each of the example projects, right-click the **References** item, then click **Add Reference**.
4. Click the **Browse** button, select WORD.DLL from the list of .dll files, then click the **Open** button.
5. Click the **OK** button to close the reference browser, and click the **OK** button for the popup window telling you that the directory WORD.DLL was found in will be added to the reference path.
6. If you have installed Microsoft Office XP, add OFFICEXP to the list of compilation constants. For each of the example projects, right-click the project root, then click **Properties**.

   The **Property Pages** dialog box appears.

7. Select **Configuration Properties**, then **Build**, then add **OFFICEXP** to the list of **Conditional Compilation Constants**. Click the **OK** button.
8. From the **Build** menu, click **Build Solution**.

To run the automation examples

1. Right-click Example1 in **Solution Explorer** and select **Set as Startup Project** from the shortcut menu, then build and run the project.

   This example shows you how to instantiate the Word object without any documents.

2. Right-click Example2 in **Solution Explorer** and select **Set as Startup Project** from the shortcut menu, then build and run the project.

   This example shows you how to create a new Word document and put some text into it.

3. Right-click Example3 in **Solution Explorer** and select **Set as Startup Project** from the shortcut menu, then build and run the project.

   This example shows you how to open an existing Word document and modify it.

4. Right-click Example4 in **Solution Explorer** and select **Set as Startup Project** from the shortcut menu, then build and run the project.

   This example shows you how to hook various events in Word.

Classes and Keywords in the Sample

This sample demonstrates the following classes:

- **System.Diagnostics.Process**
- **System.Environment**

- **System.Reflection.Missing**
- **System.Text.StringBuilder**
- **System.Windows.Forms.MessageBox**
- **System.Thread.Sleep**
- **System.Reflection.Missing.Value**
- **Microsoft.Win32.RegistryKey**
- **Word.Application**
- **Word.ApplicationEvents2_DocumentChangeEventHandler**
- **Word.ApplicationEvents2_DocumentOpenEventHandler**
- **Word.ApplicationEvents2_NewDocumentEventHandler**
- **Word.ApplicationEvents2_StartupEventHandler**
- **Word.Documents**
- **Word._Document**
- **Word.Range**
- **Word.Words**

See Also

General Samples

CarsSelector Sample (Creating and Using ASP.NET Web Pages)

This sample shows how to build a Web Forms page for selecting cars that simulates a simple e-commerce page. The sample demonstrates the following features:

- Data binding: In this sample you use more than one type of data binding:
  - Simple data binding.
  - Binding to a data table created programmatically (the data source is not coming from a database).
- Using the event handler to dynamically change the binding of some controls.
- Using validation controls.
- Using the Calendar control and validating its entries without using the validation controls.

Requirements

To build and run this sample, you need a computer running Internet Information Services 5.0 with FrontPage Server Extensions installed.

Building and Running the Sample

To create the solution

1. Copy the sample files to your hard disk.
2. Create a new **Web Application** and name it CarsSelector.
3. Delete the default Web Forms page by right-clicking the file **WebForm1.aspx**, and then click **Delete** on the shortcut menu.
4. Add the file CarsSelector.aspx to the project. To do that, right-click the project name, point to **Add** on the shortcut menu, and click **Add Existing Item**. Change the **Types of files** to All Files (*.*), then browse to the file CarsSelector.aspx, select it, and click **Open**.
5. Add the file CarsSelector.aspx.jsl to the project. To do that, right-click the project name, point to **Add** on the shortcut menu, and click **Add Existing Item**. Then browse to the file CarsSelector.aspx.jsl, select it, and click **Open**. Click **OK** to overwrite the existing file.
6. Set the Web Forms page as a start page. To do that, right-click the file **CarsSelector.aspx** and click **Set As Start Page** on the shortcut menu.

For more information on creating Web Forms applications**,** see Creating and Managing Web Forms.

To build and run this sample

1. On the **Build** menu, click **Build Solution**.
2. On the **Debug** menu, click **Start Without Debugging**.

   The Web Forms page appears. You can select a car and make choices, and then you see the result bound to a label.

To use the Web form

1. Start by selecting a car brand from the **Brand** drop-down list.
2. Choose a model from the **Model** drop-down list. You will notice that the contents of this list depend on the selected brand of the car.
3. Choose the **Year**, **Color**, and **Mileage** from the appropriate drop-down lists.
4. Enter an e-mail address in the **Your Email** text box.
5. Select one or more check boxes from the **Features Desired**.
6. Select a valid date in the **Desired Available Date** calendar.
7. Click **Submit**.

Classes and Keywords in the Sample

This sample demonstrates the following class and keywords:

- **System.Web.UI.Page**
- **DropDownList**
- **AutoPostback**

- **CheckBoxList**
- **RequiredFieldValidator**
- **RegularExpressionValidator**
- **Dilatable**

See Also

General Samples

LocalizedHelloWorld Sample (Satellite Assemblies for Console Applications)

This sample demonstrates creating and using satellite assemblies for console applications. It shows how to create resources for Visual J# console applications, localize the resources, and build localized satellite assemblies.

Although this sample was written using Visual J#, the sample principles and techniques can be used to localize any managed application written in any language.

This sample is divided into the following projects and assemblies:

Resources C++ Utility Project

This is the first project. It builds the managed resources with a custom build step that calls ResGen.exe to compile the .txt files into .resources files. The resource text file contains lines of <name>=<value> statements and comment lines that begin with a pound sign (#). In code, the strings will be loaded using the name given in this file.

Notice that for the Korean resources the file must be saved as UTF-8 to preserve the non-ASCII characters. Also note that individual cultures are not required to have a value for every name.

If a value doesn't exist for the proper culture, the resource manager automatically falls back to the parent cultures. For example, if the current culture is ko-KR, the resource manager will first look for the name in the "ko-KR" resources, then in the "ko" resource, and then finally in the default resources. Technically these resource files can be named anything, but typically they are named with the following convention:

<basename>[.<culture>].resources

If the culture is empty, it is assumed to be the default culture.

HelloWorld Visual J# Console Project

This is the second project. It builds the main Visual J# application and embeds the default resources (MyResources.resources built from MyResources.txt by the first project).

SatelliteAssemblies C++ Utility Project

This is the third project. It builds the two satellite assemblies that contain the localized resources with custom build steps. It calls Al.exe to create the satellite assembly, passing in the resource file with the "/embed" option, setting the culture with the "/culture" option, and copying the assembly information (title, version, and so on) from the main assembly with the "/template" option.

Note that the satellite assemblies must be in a subdirectory named after their culture of the main assembly, and the satellite assembly's file name is the same as the main assembly, but with ".resources" inserted before the file extension. For example, if the main assembly was "c:\bin\a.dll", the en-US satellite assembly would be "c:\bin\en-US\a.resources.dll".

Also note the second argument to the "/embed" option, the resources name. This name is the name that the resource manager will use to find the given resource group, and the name must include the proper culture string (case-sensitive) after the name and before ".resources". Because Al.exe gets most of the information from the main assembly, this third project must be built after the second one.

Building and Running the Sample

To build and run this sample

1. Open the solution LocalizedHelloWorld.sln.
2. From the **Build** menu, click **Build Solution**.
3. From the **Debug** menu, click **Start Without Debugging**.
4. Type in a culture supported by the application to see the resources in that culture (the sample ships with **en**, **en-US**, **ko-KR**, and **default** resources which are in English).

To add a new culture

The following procedure adds resources for English (United Kingdom).

1. Add a new text file named "MyResources.en-GB.txt" to the Resources C++ utility project.
2. Add the appropriate *<name>=<value>* lines to the new text file along with any desired comments.
3. Add a custom build step for the new file (this should be the same as for the other resource files in the project): `ResGen.exe "$(InputPath)"`.
4. Set the output of the custom build step to `$(InputName).resources`.
5. Build this project (no need to rebuild the main assembly).
6. Add an existing file named "..\Resources\MyResources.en-GB.resources" to the SatelliteAssemblies C++ utility project.
7. Add a custom build step for this file (this should be similar to the other files' custom builds steps):

   mkdir $(ProjectDir)..\bin\$(OutDir)\en-GB

   al.exe /culture:en-GB /out:"$(ProjectDir)..\bin\$(OutDir)\en-GB\HelloWorld.resources.dll" /embed:"$(InputPath)","HelloWorld.Resources.$(InputFileName)" /template:"$(ProjectDir)..\obj\$(IntDir)\HelloWorld.exe"
8. Set the output of the custom build step to "$(ProjectDir)..\bin\$(OutDir)\en-GB\HelloWorld.resources.dll".
9. Build this project (no need to rebuild the main assembly).
10. Run the sample and try your new resource.

To build this sample from the command line

1. Compile the resource for all cultures (from the Resources subdirectory):
   - ResGen.exe MyResources.txt
   - ResGen.exe MyResources.en-US.txt
   - ResGen.exe MyResources.ko-KR.txt
2. Compile the main assembly and embed the default resources (from the root project directory):

vjc.exe /out:bin\Debug\HelloWorld.exe /res:Resources\MyResources.resources,HelloWorld.Resources.MyResources.resources HelloWorld.jsl AssemblyInfo.jsl

3. Build the satellite assemblies (from the root project directory):

   mkdir bin\Debug\en-US

   mkdir bin\Debug\ko-KR

   al.exe /culture:en-US /out:bin\Debug\en-US\HelloWorld.resources.dll /embed:Resources\MyResources.en-US.txt,HelloWorld.Resources.MyResources.en-US.resources /template:bin\Debug\HelloWorld.exe

   al.exe /culture:ko-KR /out:bin\Debug\ko-KR\HelloWorld.resources.dll /embed:Resources\MyResources.ko-KR.txt,HelloWorld.Resources.MyResources.ko-KR.resources /template:bin\Debug\HelloWorld.exe

   Note You can add options to the Visual J# command-line for debugging or other reasons. You can also change the output directory from "bin\Debug" to a different directory.

See Also

General Samples

MultilingualForm Sample (Create and Localize Windows Forms)

This sample demonstrates creating and localizing windows forms. It shows how to use the Windows Forms designer to create, edit, and build localized applications.

For more information about localized applications, see Globalizing and Localizing.

Building and Running the Sample

To build and run this sample

1. Open the solution MultilingualForm.sln.
2. From the **Build** menu, click **Build Solution**.
3. From the **Debug** menu, click **Start Without Debugging**.
4. To see a different culture's resources, type a valid RFC 1766 culture name into the edit box and click the **switch** button. (This sample only comes with resources for "en-US" and "ko-KR".)

To localize your own Windows Forms or Web Forms

1. Select the form to be localized in the appropriate forms designer.
2. Set the form's **Localizable** property to true.
3. For each language you want to localize to, do the following:
   a. Set the form's **Language** property to the desired language.
   b. Enter the new text in the new language for each control.

   You can also change the appearance of the form's controls (size, position, and so on) for each language.

See Also

General Samples

PerfMon Sample (Monitor System Performance Using Performance Counters)

This sample is an introduction to performance counters. It demonstrates how to monitor local or remote system performance. It shows how to dynamically create timers, context menus, and graphic displays. It also demonstrates the following activities:

- Including new Visual J# classes derived from other components.
- Dynamically adding and removing event handlers and components.
- Using the **TreeView** control, **TreeNode** class, and **Timer** component.
- Using the **PerformanceCounterCategory** and **PerformanceCounter** components.

For more information on performance counters, see PerformanceCounter Class.

> **Note** This sample is functional only on Windows NT 4.0 and Windows 2000 and later.

Building and Running the Sample

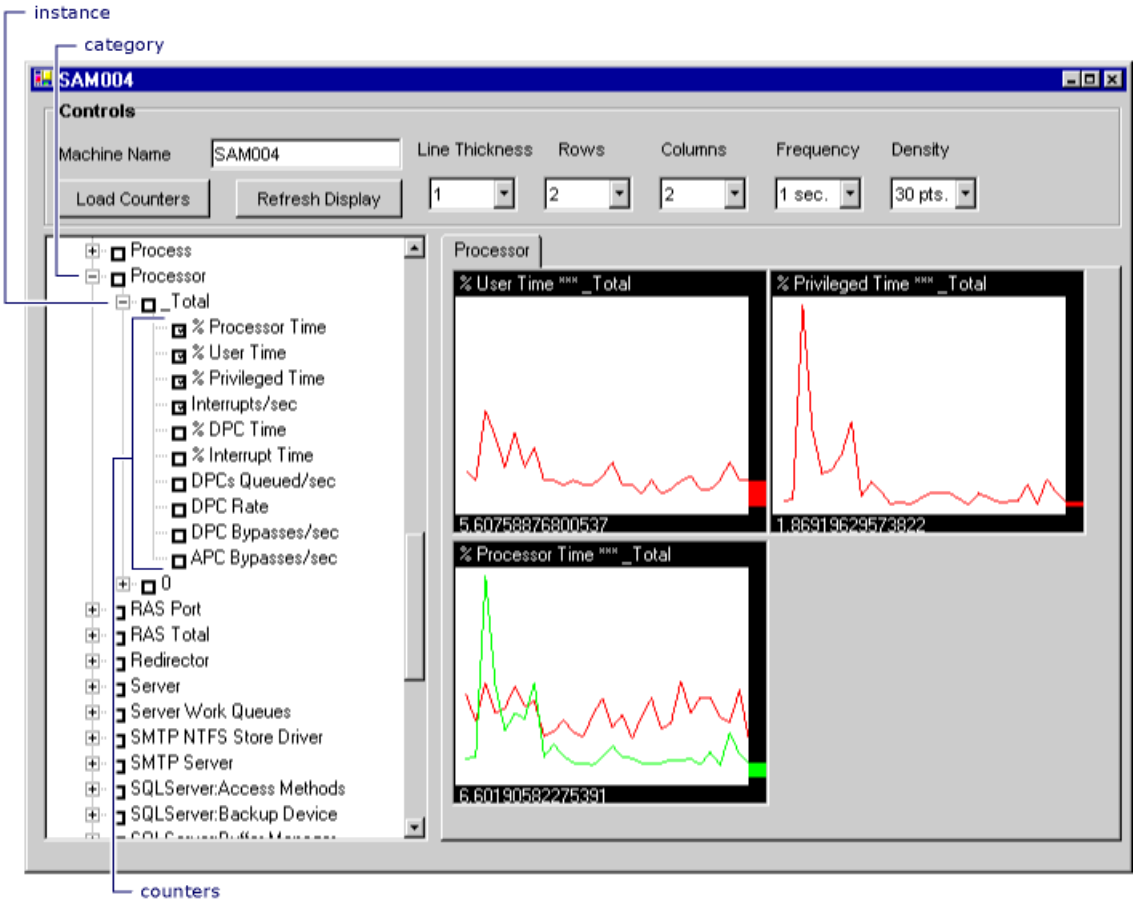
To build and run this sample

1. Open the solution PerfMon.sln.
2. From the **Build** menu, click **Build Solution**.
3. From the **Debug** menu, click **Start Without Debugging**.

   The main application window will appear.

The following figure shows the performance monitor program running. Refer to the figure, as needed, while you go through the various performance monitoring procedures.

Performance Monitor



Display Panel Properties

Each graphic display in the figure shows the following:

- The top of the display shows the name of the counter and the name of the instance separated by "***". If the category does not contain an instance name (for example, the ICMP category), the "***" is followed by a blank space.
- The bottom of the display shows the current counter value. This value may change periodically (the default is every second).
- The right side of the display shows a red bar. The height of the bar depends on the last value monitored on the counter.
- The graph plotted in the display shows the performance history for that counter.

To load the performance counters of a system

This application is an MDI application. It starts with four windows that can be used to monitor four computers. You can open as many windows as you want by clicking the **New PerfMon** command on the **Options** menu. In the following procedures, only one window is used. Therefore, you can close all of the windows except one.

1. In the **Machine Name** field, type the name of the system you want to monitor. If you do not type anything, the local machine will be monitored.
2. Click the **Load Counters** button (or press ALT+L). The application will load **PerformanceCounterCategory**, **InstanceName**, and **PerformanceCounter** components on the specified machine. The **Load Counters** button will be disabled until all of the counters on the system have been loaded.

   The tree view in the left pane will contain all of the performance categories and the performance counters available. Each leaf in the tree represents a performance counter. You can monitor the completion progress by watching the progress indicator on the bottom of the application window. After the tree has been completed, the **Load Counters** button will be enabled again.

   > **Note** To monitor a specific machine, you need administrator rights to the machine; otherwise, the **PerformanceCounterCategory** components will not be able to load the categories (security protection).

To monitor the performance of a system

1. Load the performance counters by performing the preceding steps.
2. Select one of the counters by expanding the corresponding category node. For example, to watch the performance of the processor, expand the **Processor** node, and then select one of the **Processor** instances (for example, "**_Total**" or "**0**"). Expand the instance node as well to see the counters.
3. On the leaf level, the counters are represented by check boxes. Select one of the counters, for example, **%Processor Time**. A tab, which has the name **Processor**, will appear in the right pane, and a graphic display of the selected counter will appear on the tab page.
4. To remove a specific graphic display from the tab page, clear the check box for the corresponding counter. When all of the counters in a specific page are cleared, the corresponding tab disappears.

To manage the displays properties

> **Note** Each time you change any of the display properties, click the **Refresh Display** button to apply the changes. The changes will apply for current or future displays.

- To change the line thickness, select the thickness you want from the **Line Thickness** box.
- To change the number of rows/display, select the number you want from the **Rows** box. For example, selecting one row makes the display occupy the full height of the tab page.
- To change the number of columns/display, select the number you want from the **Columns** box. For example, selecting one column makes the display occupy the full width of the tab page.

  > **Note** If the number of graphic displays exceeds the visible area of the tab page, scroll bars will be created automatically.

- To change the time interval, select the number of seconds from the **Frequency** box.
- To change the number of points to display in the graph, select a number from the **Density** box.

To merge two or more displays

1. Drag the display you want to move and drop it on the destination display. Notice that the background of the destination display becomes black when it is selected and is ready to receive the moved display. Both displays will merge, and two different graphic colors will appear (red and green in this case). For example, if you drop "**%Processor Time *** _total**" on "**%User Time *** _total**", the green color will represent the first and the red color will represent the second. The data shown in the display's top, bottom, and right side will only represent the **"%Processor Time *** _total**" counter.

2. To switch from one graph to another after a display merge, right-click the display and select the name of the graph from the shortcut menu. In the preceding example, if you want to change the display data from "**% Processor Time *** _total**" to "**%User Time *** _total**", right-click the display and select **"%User Time *** _total**" from the shortcut menu. The display will show the data of the selected counter, and the bar color on the right side will change to red (or the color representing the selected counter).
3. If you have more than two merged counters, each counter will be represented by a different color. To select the data view for any one of the counters, perform the preceding step.

To separate two or more merged displays

1. Locate the counter's check box of the display you want to separate.
2. Clear the counter's check box. The corresponding display will be removed.
3. Select the counter's check box again. A new display will appear on the tab page.

To use menu options

- To stop the graphics plotting:

  1. Click the **Option** menu (or press ALT+O).
  2. Click **Stop Graphics**.

  The graphics will freeze and no more lines will be plotted.

- To restart the graphics plotting:

  1. Click the **Option** menu.
  2. Click **Start Graphics**.

  The displays will continue plotting from the last position.

- To exit the application:

  1. Click the **Option** menu.
  2. Click **Exit**.

See Also

General Samples

ProcessController Sample (Processes and Services Monitoring)

This sample shows how to build a Visual J# Windows application for processes and services monitoring. The sample also demonstrates the following activities:

- Dynamically adding and removing event handlers.
- Using the **Process** component to:
  - Read the running processes on the local or remote machine.
  - Start a process on the local machine.
  - Stop a process on the local or remote machine.
  - Read the modules used by some running process.
  - Read all of the characteristics of some running process.
- Using the **ServiceController** component to:
  - Read the running services and device drivers on the local or remote machine.
  - Start, stop, or pause a service on the local or remote machine.
  - Check properties of the services or device drivers on the local or remote machine.
- Handling security issues related to the use of the **Process** and the **ServiceController** components.

  Note This sample runs only on the Windows NT or Windows 2000 operating systems because the **ServiceController** component is only supported on Windows NT platforms.

Building and Running the Sample

To build and run this sample

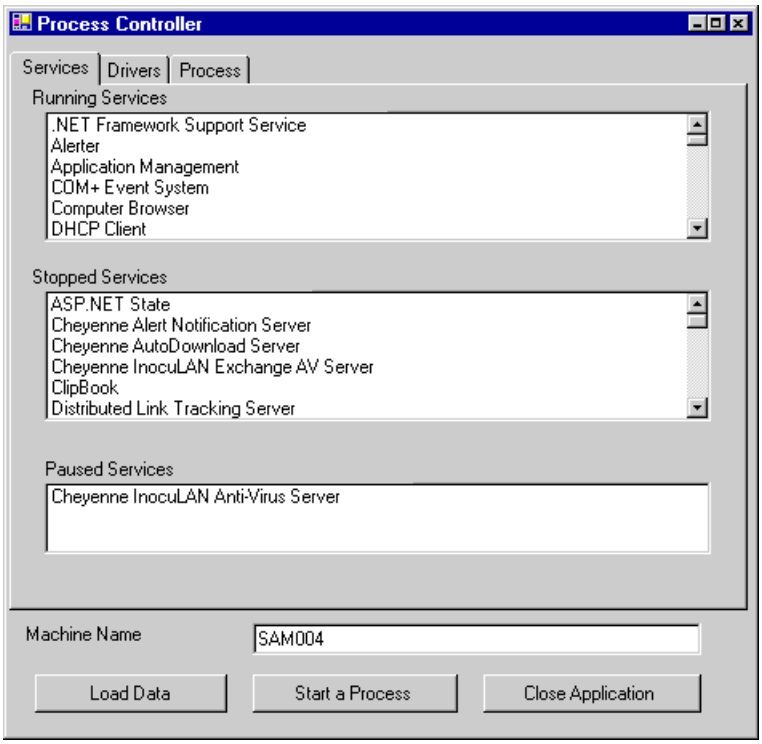
1. Open the solution ProcessController.sln.
2. From the **Build** menu, click **Build Solution**.
3. From the **Debug** menu, click **Start Without Debugging**.

   The main application window will appear.

The following figure shows the ProcessController program running. Refer to the figure, as needed, while you perform the following test procedures:

- Load all services, device drivers, and processes running on some machine in the network.
- Start, pause, or stop a service or a device driver running on the local or remote machine.
- Start a process running on the local machine.
- Terminate a process running on the local or remote machine.

Process Controller



To load all of the services, device drivers, and processes

1. Type the name of the machine you want in the **Machine Name** box.
2. Click the **Load Data** button (or press ALT+L).

To start a service

1. Click a service name in the **Stopped Services** list.
2. Right-click the service and click **Start Service** from the shortcut menu.

   After the service starts, its name should appear in the topmost list labeled **Running Services**.

   Note You cannot start a service if you do not have the administrator rights on that machine.

To stop a service

1. Click a service name in the **Running Services** list.
2. Right-click the service and click **Stop Service** from the shortcut menu.

   After the service has been stopped, the name should appear in the **Stopped Services** list.

   Note Some services do not accept the **Stopped** state. Also, you cannot stop a service if you do not have the administrator rights on that machine.

To pause a running service

1. Click a service name in the **Running Services** list.
2. Right-click the service and click **Pause** from the shortcut menu.

   If the service is paused, the service name should appear in the **Paused Services** list at the bottom of the window.

   Note Some services do not accept the **Paused** state. Also, you cannot pause a service if you do not have the administrator rights on that machine.

To check some of the service's information

1. Click a service name in any of the three lists.
2. Right-click the service and click **Show Service Info** from the shortcut menu. A new window containing the service information will appear.

To verify and manage device drivers

1. Click the **Drivers** tab at the top of the window.

2. Go through the preceding steps to start, stop, and pause a service. Exactly the same rules mentioned for services apply to the device drivers.

To verify and manage processes

1. Click the **Process** tab at the top of the window.
2. Select a process name in the left-most list. At this point, information about that service should appear in the list on the right.

   Note If you are not an administrator on the target machine, you will not be able to see or manage any process.

To start a process

1. Click the **Start a Process** button (or press ALT+S).
2. Select the process you want to start in the **Open** dialog box.
3. Click the **Open** button in the dialog box. If your target machine is the local machine, the process name should appear in the leftmost list.

   Note If the target machine is a remote machine, the process will start on the local machine, by default, if possible.

To stop a process

1. Click a process in the left-most list.
2. Right-click the process and click **Terminate Process** from the shortcut menu.

   After the process has been terminated, its name should disappear from the list.

To load the services, drivers, and processes from another machine

1. Type the name of the machine in the **Machine Name** box.
2. Click the **Load Data** button (or press ALT+L).

To stop the ProcessController application

- Click the **Close Application** button (or press ALT+C).

  After you click the button, the window will slowly fade away. This effect is obtained by handling the window's **Opacity** property.

Classes in the Sample

The following classes are used in this sample.

File	Description
MainForm.jsl	Describes the **MainForm** class, which is a **System.Windows.Forms.Form** type class. Its designer contains most of the UI controls needed for this application.
ProcessControllerManager.jsl	Contains the implementation of the **ProcessControllerManager** class. This class handles processes management.
DriverControllerManager.jsl	Contains the implementation of the **DriverControllerManager** class. This class is in charge of managing the device drivers on a given machine.
ServiceControllerManager.jsl	Contains the implementation of the **ServiceControllerManager** class. This class describes the logic to control the services on a given machine.
ServiceInfo.jsl	Implements the **ServiceInfo** class, which is a **System.Windows.Forms.Form** class and is used to display services and drivers information only.

See Also

General Samples

Scribble Sample (MDI Drawing Application)

This sample shows how to develop a Windows Forms MDI application using Visual J# and the .NET Framework classes. Scribble is a small drawing application that lets you draw free-hand drawings using the mouse and save the images in a file.

Building and Running the Sample

To build and run this sample

1. Open the solution Scribble.sln.
2. From the **Build** menu, click **Build Solution**.
3. From the **Debug** menu, click **Start Without Debugging**.

When you run the sample, you get a drawing surface and the following menu items:

- **File** menu for input and output operations.
- **Edit** menu to cut, copy, and paste.
- **Pen** menu to select the pen thickness.
- **View** menu to show or hide the toolbar and the status bar.
- **Window** menu to create, tile, or cascade windows.
- **Help** menu to display the help topics and the About Scribble text box.

In addition to the menu options, the user interface includes a toolbar with buttons to duplicate the menu commands: **New**, **Open**, **Save**, **Print Preview**, **Print**, and **Help**.

Other Versions of Scribble

Scribble is also available as an MFC sample, as a Managed Extensions for C++ sample, and as Visual Basic sample:

- Scribble Sample: MFC MDI Drawing Application
- Scribble Sample: MDI Drawing Application Using Managed Extensions for C++
- Windows Forms: Scribble Sample

See Also

General Samples

Technology Samples

The samples in this section illustrate the various technologies and features in the .NET Framework and in Visual J#. The intent is to instruct on a single technology or a few related technologies. These samples are not strictly useful as complete applications and tend to exhibit limited error handling and object-oriented development:

Category	Sample
Class loader	ClassLoader Sample (Modify Custom Class Loader for Assemblies)
CLS Extender	CLSExtender Sample (Authoring .NET Framework Properties, Events and Delegates)
CodeDOM	CodeDOM Sample (Use CodeDomProvider)
Cross language	CrossLanguage Sample (Visual J# Interop with Other Managed Languages)
Dynamic loading	DynamicLoading Sample (Load Classes from Deployed Assemblies)
Java-language/COM	COMCallsJava Sample (Upgrading Visual J++ 6.0 Java-Language Components)
	JavaCallsCOM Sample (Upgrading Visual J++ 6.0 Java-Language Components)
Java-language/COM (using .NET Framework semantics)	COMCallsJava_NET Sample (Expose Managed Components to COM)
	JavaCallsCOM_NET Sample (Expose COM Objects to Managed Components)
Platform invocation	PInvoke Sample (Call Unmanaged APIs)
Scoping	Scoping Sample (Using /securescoping)
Security providers	SecurityProviders Sample (Specify Third Party Security Providers)
Using .NET Framework	Attributes Sample (Use .NET Framework Attributes)
	Enums Sample (Use Enumerations)
	EventsAndDelegates Sample (Use .NET Framework Events and Delegates)
	Minesweeper Sample (Build a Version of the Minesweeper Game)
	Properties Sample (Use .NET Framework Properties)
	ValueTypes Sample (Use .NET Framework Value Types)
Remoting service	Remoting Sample
Windows Forms and extending WFC application	WFCUsingWinForm Sample (Extend WFC Application Built in Visual J++ 6.0 Using Windows Forms)

See Also

Visual J# Samples | Application Samples | General Samples | Cross-Language Samples

ClassLoader Sample (Modify Custom Class Loader for Assemblies)

This sample shows how custom class loaders in existing Java-language applications can be modified to load classes from managed assemblies. For new code, it is recommended that classes be searched and loaded from managed assemblies using common language runtime semantics and .NET Framework APIs rather than using custom class loaders. Class loaders are only intended to be used by existing Java-language applications. The **resolveClass** and **defineClass** methods in the class **java.lang.ClassLoader** are unsupported. In existing Java-language applications, the **loadClass** method will have to be rewritten to call one of the versions of **Class.forName** and then make appropriate entries in the application configuration (.config) file so that the managed assemblies needed by the application can be located and loaded. If the managed assemblies are located in the application's working directory, then no configuration file is required and the **Class.forName** search heuristics will locate the class. Please refer to Application Configuration Files for information on application configuration (.config) file syntax.

Building and Running the Sample

To build this sample

- From the development environment, open the ClassLoader.vjsproj file and build the sample (CTRL+SHIFT+B).

  -or-

- From the command line, type **BUILD.bat**.

To run this sample:

- From the development environment, press F5 to build and run the sample.

  -or-

- From the command line, type **ClassLoaderSample.exe** from the command line.

Command-Line Requirements

For the Visual J# samples to function properly when building and running from the command line, you need to run vsvars.bat (located in the <%Program Files%>\Microsoft Visual Studio .NET 2003\Common7\Tools directory). Alternatively, you can also use the Visual Studio .NET command prompt (click **Start**, then Programs, then **Microsoft Visual Studio .NET 2003**, and finally **Visual Studio .NET Tools**).

For the Visual J# samples to function properly when building and running from the command line, you must additionally set the **Path** variable to include the <*Microsoft Visual J++ 6.0 install root*> directory.

See Also

Technology Samples

CrossLanguage Sample (Visual J# Interop with Other Managed Languages)

This sample shows how Java-language syntax can be used in Visual J# to access classes written in other languages and how other languages can access Visual J# classes. Also shown is how other languages can catch Java-language exceptions thrown by components written in Java-language. This sample creates three assemblies. These three library or DLL assemblies define simple classes written in Managed Extensions for C++, Visual Basic, and Visual J#. These classes derive from each other and ultimately from the base class written in Managed Extensions for C++. Finally, the executable written in C# creates instances of the Visual J# class and calls methods on it. The .NET Framework is an environment where various developers can work together seamlessly while developing in their language of choice.

Building and Running the Sample

To build this sample

- Type **BUILD.bat** from the command line.

  Note The location of one of the DLLs needed to compile the sample is hard-coded in the build script to C:\Windows\Microsoft .NET\Framework\v<%version%>\vjslib.dll. This may need to be changed based on the version of Visual J# installed on the machine and the name of the Windows directory.

To run this sample

- Type **CrossLang.exe** from the command line.

For each object instantiated, the code called was written in at least two different development languages.

Command-Line Requirements

For the Visual J# samples to function properly when building and running from the command line, you need to run vsvars.bat (located in the <%Program Files%>\Microsoft Visual Studio .NET 2003\Common7\Tools directory). Alternatively, you can also use the Visual Studio .NET command prompt (click **Start**, then **Programs**, then **Microsoft Visual Studio .NET 2003**, and finally **Visual Studio .NET Tools**).

For the Visual J# samples to function properly when building and running from the command line, you must additionally set the **Path** variable to include the <*Microsoft Visual J++ 6.0 install root*> directory.

See Also

Technology Samples

Enums Sample (Use Enumerations)

This sample demonstrates the use of .NET Framework enumerations. There is support to consume .NET Framework enumerations from Visual J#, but there is no support to author (define) enumerations.

Building and Running the Sample

To build this sample

- From the development environment, open the Enums.vjsproj file and build the sample (CTRL+SHIFT+B).

  -or-

- From the command line, type **BUILD.bat**.

To run this sample

- From the development environment, press F5.

  -or-

- From the command line, type **EnumSample.exe**.

Command-Line Requirements

For the Visual J# samples to function properly when building and running from the command line, you need to run vsvars.bat (located in the <%Program Files%>\Microsoft Visual Studio .NET 2003\Common7\Tools directory). Alternatively, you can also use the Visual Studio .NET command prompt (click **Start**, then **Programs**, then **Microsoft Visual Studio .NET 2003**, and finally **Visual Studio .NET Tools**).

For the Visual J# samples to function properly when building and running from the command line, you must additionally set the **Path** variable to include the <*Microsoft Visual J++ 6.0 install root*> directory.

For more information on consuming .NET Framework enumerations, see Using Enumerations.

See Also

Technology Samples

Properties Sample (Use .NET Framework Properties)

This sample demonstrates the use of properties supported by the.NET Framework. There is support to get and set properties using the get_ and set_ property accessor methods.

Building and Running the Sample

To build this sample

- From the development environment, open the Properties.vjsproj file and build the sample (CTRL+SHIFT+B).

  -or-

- From the command line, type **BUILD.bat**.

To run this sample:

- From the development environment, press F5.

  -or-

- From the command line, type **Properties.exe**.

Command-Line Requirements

For the Visual J# samples to function properly when building and running from the command line, you need to run vsvars.bat (located in the <%Program Files%>\Microsoft Visual Studio .NET 2003\Common7\Tools directory). Alternatively, you can also use the Visual Studio .NET command prompt (click **Start**, then **Programs**, then **Microsoft Visual Studio .NET 2003**, and finally **Visual Studio .NET Tools**).

For the Visual J# samples to function properly when building and running from the command line, you must additionally set the **Path** variable to include the <*Microsoft Visual J++ 6.0 install root*> directory.

See Also

Technology Samples

ValueTypes Sample (Use .NET Framework Value Types)

This sample demonstrates the use of value types supported by the .NET Framework. Visual J# does not support authoring (defining) value types. However, there is support to use the value types defined in .NET Framework class libraries.

Building and Running the Sample

To build this sample

- From the development environment, open the ValueTypes.vjsproj file and build the sample (CTRL+SHIFT+B).

  -or-

- From the command line, type **BUILD.bat**.

To run this sample:

- From the development environment, press F5.

  -or-

- From the command line, type **ValueTypes.exe**.

Command-Line Requirements

For the Visual J# samples to function properly when building and running from the command line, you need to run vsvars.bat (located in the <%Program Files%>\Microsoft Visual Studio .NET 2003\Common7\Tools directory). Alternatively, you can also use the Visual Studio .NET command prompt (click **Start**, then **Programs**, then **Microsoft Visual Studio .NET 2003**, and finally **Visual Studio .NET Tools**).

For the Visual J# samples to function properly when building and running from the command line, you must additionally set the **Path** variable to include the <*Microsoft Visual J++ 6.0 install root*> directory.

See Also

Technology Samples

WFCUsingWinForm Sample (Extend WFC Application Built in Visual J++ 6.0 Using Windows Forms)

This sample demonstrates how a WFC application built in Visual J++ 6.0 can be extended using Windows Forms. A simple WFC application built in Visual J++ 6.0 is upgraded to Visual J#. This sample is extended to launch a simple Windows Forms component on clicking a button or selecting a menu option. A similar approach can be used to invoke and launch any Windows Forms component from a WFC application.

Building and Running the Sample

To build this sample

- From the IDE, open WFCUsingWinForm.sln and build the sample (CTRL+SHIFT+B).

  -or-

- From the command line, type **BUILD.bat**.

To run this sample

- From the development environment, press F5.

  -or-

- From the command line, type **WFCUsingWinForm.exe**.

To use the WFC application

1. Click the button to launch a Windows Forms component.
2. From **File**, choose **Start Win Form** to launch the same Windows Forms component.

Command-Line Requirements

For the Visual J# samples to function properly when building and running from the command line, you need to run vsvars.bat (located in the <%Program Files%>\Microsoft Visual Studio .NET 2003\Common7\Tools directory). Alternatively, you can also use the Visual Studio .NET command prompt (click **Start**, then **Programs**, then **Microsoft Visual Studio .NET 2003**, and finally **Visual Studio .NET Tools**).

For the Visual J# samples to function properly when building and running from the command line, you must additionally set the **Path** variable to include the <*Microsoft Visual J++ 6.0 install root*> directory.

See Also

Technology Samples